Aalto University
School of Science
Master's Programme in Computer, Communication and Information Sciences

Rohan Chauhan

Elicitation of Non-Linearity from Expert Drawing

Master's Thesis
Espoo, May 22, 2019

Supervisor: Professor Samuel Kaski
Advisor: Tomi Peltola D.Sc. (Tech.)

Aalto University
School of Science
Master's Programme in Computer, Communication and Information Sciences

ABSTRACT OF MASTER'S THESIS

Author: Rohan Chauhan

Title:
Elicitation of Non-Linearity from Expert Drawing

Date: May 22, 2019 **Pages:** 57

Supervisor: Professor Samuel Kaski

Advisor: Tomi Peltola D.Sc. (Tech.)

Machine learning methods do not perform very well with little data because there is not enough information to learn. The choice is to either obtain more data or elicit knowledge from an expert. Obtaining more data might be infeasible because of the associated cost or required time. In such cases, we opt for expert knowledge elicitation.

Current expert knowledge elicitation methods either query the user for data points or regarding the relevance of parameters. However, there is no method which allows expressing the non-linearity intuitively without requiring knowledge of Bayesian statistics. We propose expert knowledge elicitation through drawing where the expert draws the fit through data points. We then combine the observed data and drawing data to select the right kernel for a Gaussian process. We also conduct a user study for testing the usability of the proposed method. We obtain better performance with the proposed model for kernel selection and extrapolation in comparison to the baseline model using only observed data.

Keywords: expert knowledge elicitation, Gaussian processes, drawing, kernel selection

Language: English

Acknowledgements

A thesis is said to be the most crucial part of graduate school. The thesis shows the ability of a student to perform independent research. First of all, I would like to express my deepest gratitude to my supervisor Professor Samuel Kaski. Professor Kaski acted as a mentor throughout my thesis not only by posing the right questions but also helping me transition into every phase of the thesis.

Secondly, I would like to thank my advisor Dr Tomi Peltola who guided me in the right direction. Tomi provided useful feedback and suggestions to improve both the direction of the research and my thesis.

I would also like to thank my friends and family for giving me emotional support. Lastly, I would like to thank Co-Adaptation project which funded the research done during my thesis.

Espoo, May 22, 2019

Rohan Chauhan

Contents

List of Tables

List of Figures

Chapter 1

Introduction

Machine learning algorithms perform well on tasks with large datasets as seen with the application of deep learning to computer vision, speech recognition and reinforcement learning. However, small datasets still pose a challenge for most machine learning algorithms. Small datasets are present in many fields such as precision medicine, material sciences and short term forecasting. Using simple models on small datasets can lead to high bias whereas complex models on small datasets can suffer from overfitting leading to high variance in predictions. Bayesian methods can be useful for small datasets because they can account for uncertainty in their predictions; however, still, the model is mostly dependent on prior for small datasets.

For a machine learning model to learn from less data, there should be enough information to capture the hidden pattern. There are several techniques such as generating synthetic data and changing the loss function to get better predictive performance, but they usually do not work very well. In such cases, the only option is to either obtain more data or take the help of expert knowledge. Obtaining more data can sometimes be infeasible because of the associated cost or required time.

Expert knowledge elicitation is the process of collecting knowledge from an expert that is relevant to the task at hand. Experts have real-world context regarding the problem which helps them to reduce the problem by simplifying the assumptions or providing relevance feedback on features or parameters of the model. Some expert knowledge elicitation techniques require the expert to have specialised knowledge of the machine learning method which can be burdensome as well as deviate their research focus from their original goal. Several expert knowledge elicitation techniques query the user for point estimates or interval estimates. Another method of expert knowledge elicitation is active elicitation in which AI queries the expert to learn about the problem. However, active elicitation requires a large number of

steps to steer the AI as per the expert. There are no expert knowledge elicitation methods which allow an expert to express their belief regarding the fit of function through the data intuitively without diving deep into machine learning. The lack of such a method limits the ability of domain experts to apply machine learning on small datasets.

1.1 Motivation

In statistics, the fit of a function is used to define how well the learnt function models the actual function from which data is generated. Underfitting occurs when the trained model is too simple to explain the data and overfitting occurs when the trained model learns more number of parameters than required to explain the data. Figure 1.1 illustrates underfitting, good fit and overfitting for the same data.

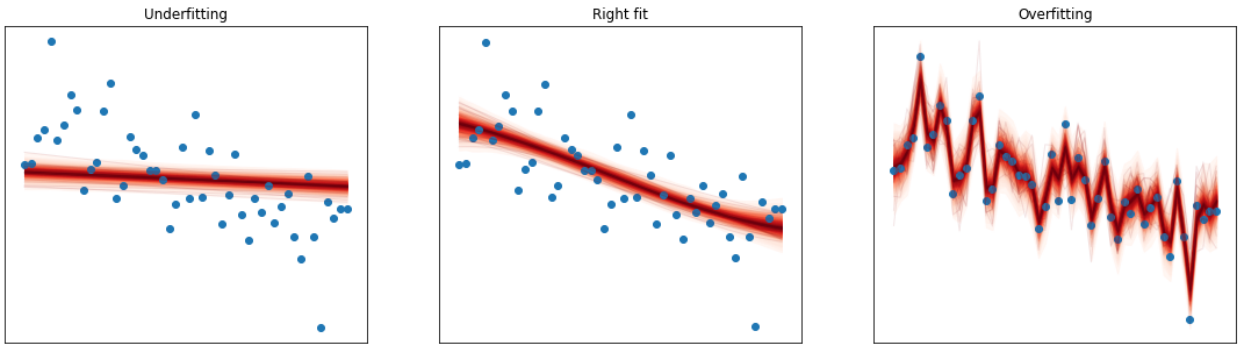

Figure 1.1: Types of fit.

Given a small dataset, how can an expert provide information about that approximate target function or the "fit" of the function? In case, the expert is well versed with machine learning, the expert can choose the appropriate algorithm depending upon the criteria. However, such models will still suffer from problems of little data. If we choose to elicit knowledge from an expert using active learning, we can query the expert to provide feedback until we learn the appropriate model. We can also choose to fit multiple models to the data and ask the expert to select the best model.

This thesis focuses on learning the right fit with the help of expert drawing as illustrated in Figure 1.2. Not only the idea of drawing seems more intuitive, but it is also better than querying point estimates regarding parameters such as lengthscale or data points itself. We combine expert drawing and the

dataset to determine the final fit of the function by selecting the appropriate kernel. A schematic representation of model is shown in Figure 1.3.

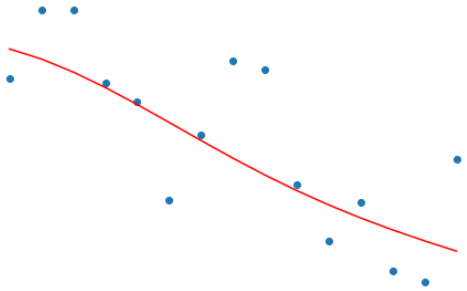

Figure 1.2: Expert drawing to provide information regarding the fit of function.

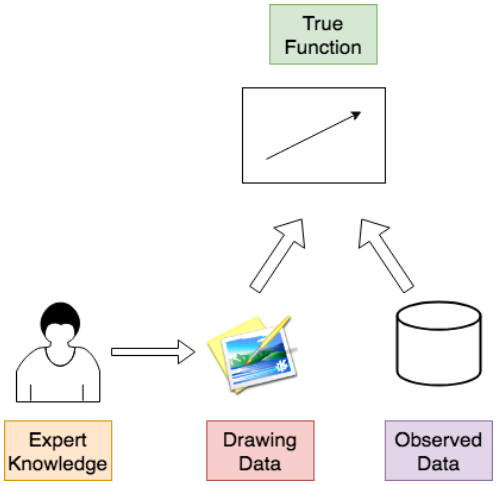

Figure 1.3: Proposed method.

1.2 Goal and Contribution of Thesis

The goal of the thesis is to elicit non-linearity from the expert through drawing. Non-linear behaviour can be explained merely as a function in which the change in output is not proportional to its input. We can model non-linear behaviour using Gaussian processes by selecting a proper covariance

function which decides the type of function and lengthscale which controls the smoothness of the functions induced by the covariance function.

The main contribution of the thesis is as follows:

- We propose a novel approach for eliciting knowledge from an expert through drawing. We also discuss other ways of acquiring expert knowledge for eliciting non-linearity and compare them to our proposed method.
- We formulate an error model of the user to capture the error during drawing.
- We also conduct a user study for usability testing of the method. For the user study, we also implement a user interface for eliciting the expert drawing.
- In simulation experiments and user study, we demonstrate better extrapolation performance with little data and expert drawing compared to the baseline model using only data.

1.3 Outline of Thesis

The rest of the thesis is structured as follows:

- Chapter 2 introduces the background knowledge required for understanding the thesis. In this chapter, we give a brief introduction to Gaussian process regression. Later on, we emphasise the importance of elicitation and analyse several approaches to the problem.
- Chapter 3 explains the research methodology followed in the thesis so that the experiments can be understood easily.
- Chapter 4 describes experiments conducted to validate our proposed method. Each experiment introduces the objective followed by the setup for the experiment. We then present our results and the analysis for the experiment.
- Chapter 5 presents a summary of the proposed methodology. We restate the contribution of this thesis and also suggest future directions for further research.

Chapter 2

Background

This chapter provides an introduction to Gaussian processes for regression and prior elicitation. We discuss prediction, hyperparameter learning and model selection in Gaussian process framework. We also briefly review methods for kernel learning.

Rest of the chapter discusses the importance of elicitation in the presence of little data. After reviewing current methods for elicitation; we examine drawing as a form of elicitation. The main contribution of this chapter is to explain the choice of drawing for elicitation from the expert.

2.1 Gaussian Process

A Gaussian process (GP) is defined such that a finite collection of random variables defined by it have a joint multivariate normal distribution [12]. Gaussian process is a stochastic process $f(x)$ specified by it's mean function $m(x)$ and it's covariance function $k(x, x')$ where

$$m(x) = \mathbb{E}[f(x)],$$
$$k(x, x') = \mathbb{E}[(f(x) - m(x))(f(x') - m(x'))].$$

The notation for Gaussian process $f(x)$ is given below:

$$f(x) \sim \mathcal{GP}(m(x)\,,\, k(x, x').$$

Usually, the mean for Gaussian process is taken as zero for notational simplicity. The behaviour of GP away from the training points can be modelled with the help of mean and covariance function.

The covariance function is also known as the kernel. The covariance function must be symmetric and positive definite. We can use the covariance

function to specify the prior over the type of functions. A combination of covariance functions can be added or multiplied to model more complex data.

2.1.1 Prediction

As per Bayes rule, the posterior distribution is given as

$$posterior = \frac{likelihood \times prior}{marginal\ likelihood}$$

or

$$p(\theta \mid y, X) = \frac{p(y \mid X, \theta)p(\theta)}{p(y \mid X)},$$

where θ represents the parameters of model and X, y is the training dataset. Likelihood is the probability density of the data points given the parameters of the model. Prior is the the belief over parameter values before we have seen the data points and the marginal likelihood is the normalizing constant given by

$$p(y \mid X) = \int p(y \mid X, \theta)p(\theta)d\theta.$$

The difference between a Bayesian linear regression and Gaussian process is that the former has prior on weights of the model and the latter has prior on function. Let y be the observations and f_* be the function values corresponding to the test set. The joint distribution for noisy model ($y = f(x) + \epsilon$) is given as:

$$p(y, f_*) = \mathcal{N}\left(0, \quad \begin{bmatrix} K_{ff} + \sigma^2 I & K_{ff_*} \\ K_{f_*f} & K_{f_*f_*} \end{bmatrix}\right),$$

where K_{ff} is the covariance matrix of the training dataset, $K_{f_*f_*}$ is the covariance matrix of the test set and K_{f_*f} is the covariance matrix between training dataset and test dataset. By conditioning over the joint distribution, we can get the predictive distribution for Gaussian process as

$$p(f_* \mid X, y, X_*) = \mathcal{N}\left(K_{f_*f}(K_{ff} + \sigma^2 I)^{-1}y), \quad K_{f_*f_*} - K_{f_*f}(K_{ff} + \sigma^2 I)^{-1}K_{f_*f}^T\right),$$

where X, y is the training data and X_* are the test points. We can use the conditioning and marginalisation properties because Gaussian process by definition is a collection of random variables forming a joint multivariate Gaussian distribution.

2.1.2 Hyperparameter Learning

We need to learn the residual variance and the hyperparameters of the selected covariance function from the training dataset. Calculating the integral of marginal likelihood for GP with non Gaussian observation model is not analytically tractable, and we have to resort to approximation or Markov chain Monte Carlo (MCMC) methods. We can also maximise the marginal likelihood together with a prior term for the hyperparameters which is known as maximum a posteriori estimate (MAP). With flat priors, MAP estimate is equivalent to maximising the likelihood of data with respect to parameters which is known as maximum likelihood estimate (MLE).

2.1.3 Model Selection

The marginal likelihood is the probability density of data given the model [12]. By calculating the probability density of data over the entire parameter space, marginal likelihood automatically trades off model complexity with data fit. For Gaussian process, the marginal likelihood is given by

$$p(y \mid X) = -\frac{1}{2} y^T K_y^{-1} y - \frac{1}{2} \log \mid K_y \mid -\frac{n}{2} \log 2\pi.$$

The first term, second term and third term in the above equation correspond to data fit term, model complexity term and constant term respectively.

2.1.4 Learning Kernels

The choice of kernel or covariance function is crucial to the performance of Gaussian process. The kernel of GP specifies the prior over the underlying function. Initially, the kernels were designed by a human expert as seen for Mauna Loa CO_2 in GPML [12]. Riihimaki et al. introduced a method for including monotonicity information in GP using virtual observations having Gaussian distribution [14]. Another way to learn expressive kernels is with the help of data itself. There are two approaches to learning expressive kernel either by learning composition of kernels or by modelling its spectral representation.

Duvenaud et al. learned a composition of kernels by greedy search over a context-free grammar of kernels [6][7]. However, the method is slow which was addressed by approximating the compositional kernels using neural network architecture in [19]. Sun et al. used the first layer to represent the base kernel family as in [7] and then the subsequent layers are based on

compositional rules. On the other hand, Wilson and Adams proposed modelling the spectral density using a Fourier transform of Gaussian mixture [22]. The spectral kernel introduced by Wilson and Adams could only approximate all stationary kernels which might not be sufficient for non-stationary data. Hence, Remes et al. introduced spectral kernel family which are non-stationary and non-monotonic [13] by solving generalised Fourier decomposition of non-stationary kernels. However, these approaches do not work well with little data.

2.2 Expert Knowledge Elicitation

The goal of elicitation is to capture the beliefs of the expert for improving the model of the observed data [8]. Bayes theorem derives the posterior distribution based on the likelihood of data given the model and the prior distribution of the parameters of a model. From a Bayesian point of view, the purpose of elicitation is to formulate the prior from expert knowledge. Elicitation can be applied to inform the relevance of features in a model or specify a distribution for parameters of the model. In the case of little data, the expert can be queried to provide a prediction for data points.

However, elicitation is not an easy task. Firstly, quantifying our beliefs about a parameter is challenging to convey numerically. Secondly, most of the time, domain experts are not familiar with statistics and probability which can make it difficult to elicit probability distributions over random variables. Nevertheless, not being able to use the available expert knowledge can decrease the expected utility of an optimal decision. Elicitation also encourages experts to think more about their modelling assumptions leading to a better model.

In this thesis, we introduce drawing as a form of elicitation. Drawing is a more natural form for describing the fit of function through a given set of observation points. To further emphasise our position, we have discussed below some other elicitation strategies which can be used to learn the fit of a function from an expert:

1. **Custom kernel design:** As already discussed earlier, the choice of covariance function in Gaussian process expresses our belief over a prior distribution of functions. Given this fact, the expert can design a custom kernel as seen in GPML [12] for Mauna Loa atmospheric CO_2. The expert can model the trend, seasonal variations and residuals by a combination of kernels. They would also need to set reasonable priors for each of the hyperparameters of the designed kernel. This method

is not only meticulous but also requires the expert to be well versed in statistics and probability.

2. **Selecting the best fit from a subset of models:** In this method, we can present the user with several models fitted with different kernels [17]. The user selects the model with the best fit and in the next iteration is again provided with several models fitted with more complex kernels. The process is repeated until the user is satisfied with the results. The drawback of this process is that it can be time-consuming for the user to evaluate all the models. It can also be possible that the final model expresses only a subset of the beliefs of the user because steering AI's behaviour is another difficult task.

3. **Querying points for model selection:** In this elicitation method, the expert answers queries for providing data points which can help us reject other models. The queries are designed such that it offers maximum information gain and help us reject other hypotheses. However, this method also requires a lot of iterations. Also, as discussed earlier, expressing function values in terms of point estimates can be difficult for an expert.

All of the methods discussed above except custom kernel design, do not allow the user to express their beliefs completely. On, the other hand, designing custom kernels can be too much to ask from a domain expert.

In the drawing, knowing the relationship between the dependent variable and the independent variable is enough. Given a set of observed data points, the expert can draw how the unknown function should vary between these points. The advantages of drawing as elicitation are as follows:

1. Drawing can help us by capturing the shape of the unknown function.

2. Drawing not only provides information regarding the shape of function but we can also learn more about the hyperparameters of the unknown function. We can jointly optimise the model with respect to observed data and drawing data.

3. In comparison to the elicitation methods discussed above, drawing can be done in one step. It doesn't require several iterations, unlike other elicitation methods to express the belief of expert.

In spite of all these advantages, even drawing might not be sufficient to express the beliefs of expert accurately. The expert might have to redraw several times to get the drawing right. A wrong drawing can lead to incorrect results which is the case with any other elicitation method.

Chapter 3

Methods

In chapter 2, we introduced the Gaussian process and talked about the benefits of expert knowledge elicitation using drawing. This chapter discusses the problems with human drawing and introduces our proposed model which tackles these problems. We introduce our proposed model and the baseline model against which we compare. We then elaborate on different choices of the criterion for kernel selection and justify our choice for using BIC.

The main contribution of this chapter is to highlight the methodology used in the thesis for expert knowledge elicitation using drawing.

3.1 Baseline Model

The baseline model is a standard GP regression model where we assume a GP prior on the underlying function with a Gaussian noise model for the observed data.

$$f(x_{obs}) \sim \mathcal{GP}_{obs}(0\,,\, k_{obs}(x, x')),$$
$$y_{obs} \sim \mathcal{N}(f(x_{obs})\,,\, \sigma_{obs} I).$$

Observed data refers to the little data which we already have while the expert data refers to the data from the expert drawing.

3.2 Problems with Human Drawing

We as humans are not able to draw accurate straight lines or smooth curves. Drawing or writing is a complicated motion involving the coordination of several joints based on visual feedback from our eyes. Hence, we need to preprocess the drawing from the user before feeding the data into the model.

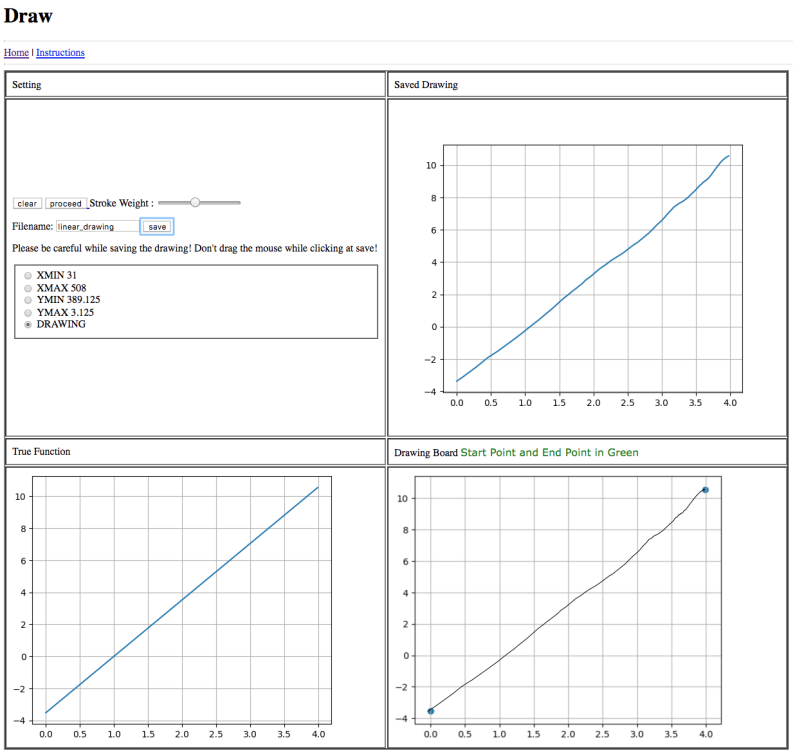

Figure 3.1: User interface.

The error which we make while drawing also varies at every position. This error is different from random error. It is a form of systematic error which keeps changing known as drift. First, we briefly discuss the user interface to explain the drawing process. Then, we discuss the preprocessing steps and explain the difference between random error and systematic error.

3.2.1 User Interface

We simulate the expert from a naive user by showing him what he/she has to draw. Before drawing, the expert has to select minimum and maximum x and y coordinate, so that they can draw on the scale of the data.

The expert sees what he has to draw on the left and draws on the right side canvas as shown in Figure 3.1. After completing the drawing, the expert saves his drawing by clicking on the save button. After clicking, the expert is shown the preprocessed drawing. They can either choose to proceed with the drawing or re-draw to improve it further. There is also a clear button for clearing the canvas.

Different people have different drawing speeds. The javascript required

for drawing runs in the background at 60 frames per second which can lead to a different number of points captured on saving the drawing. Users can also pause while drawing to re-correct what they have drawn which can also lead to a different number of points. The likelihood of expert data depends on the number of drawing points. We use linear interpolation to learn a function from the expert drawing and then generate new 500 points to approximate the drawing which helps us in comparing users and lowers computational cost for the model.

Implementation

The user interface was implemented with the help of HTML, CSS and a javascript library called p5.js [11]. Gaussian process was implemented with the help of PyMC3 which is a python library for probabilistic machine learning [16].

3.2.2 Systematic Error

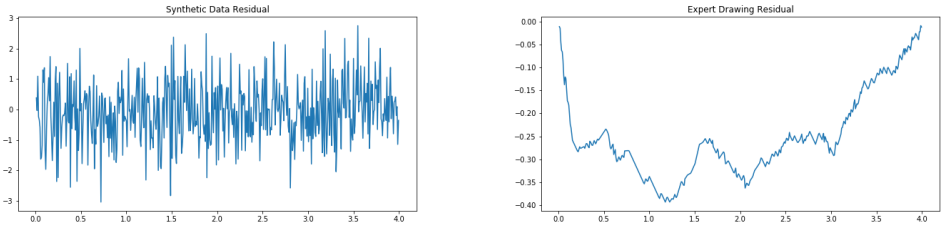

Figure 3.2: Random error vs Systematic error.

Random error as shown on the left side of Figure 3.2 are always present in measurements and can be eliminated through statistical analysis, for example, taking an average of many measurements. Systematic error, unlike random error, cannot be eliminated from statistical analysis. We can try to eliminate the cause of the systematic error by making changes in apparatus or experiment methodology such as calibration.

The error made by human drawing is a form of changing systematic error called drift. This error is different every time the user draws. Figure 3.2 shows the residuals from expert drawing when asked to draw a known function. We need to model the error in expert drawing so that we can separate the underlying latent function (signal) from noise. For modelling drift, we learn an error model for the systematic error of the drawing. We learn the error model by asking the expert to draw predefined functions during

the calibration stage and then fix the learned error model in our proposed model.

3.2.3 Error Model

The error model captures the error made by the user during drawing. We ask the user to draw on three predefined functions drawn from linear, periodic and RBF kernel. Figure 3.3 shows the three predefined function on top and the figure drawn by expert below. The user is shown the function he has to draw, and once the drawing is complete, we calculate the residuals as shown in Figure 3.4.

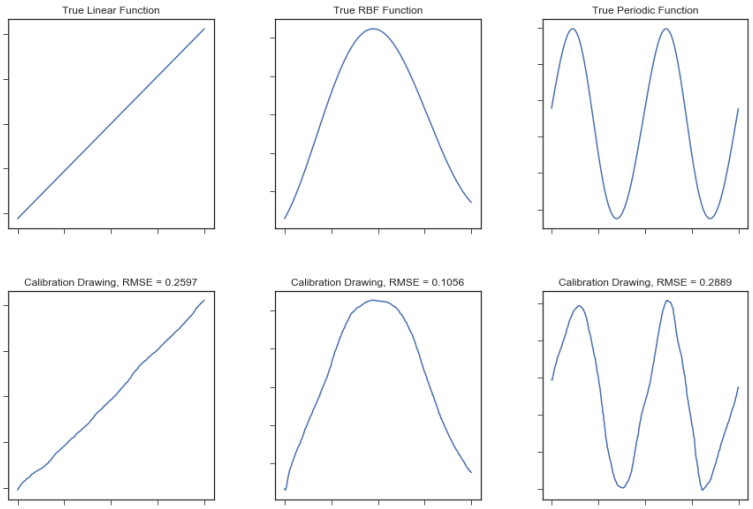

Figure 3.3: Calibration drawing by the expert.

We set a GP prior on the underlying error signal with a Gaussian noise model. We chose Matern32 kernel because the residuals are similar to functions that can be modeled by the matern kernel.

$$f(x_{error}) \sim \mathcal{GP}_{error}(0\,,\, k_{error}(x, x')),$$
$$y_{error} \sim \mathcal{N}(f(x_{error})\,,\, \sigma_{error} I).$$

After calculating the residuals for the three drawings, we learn a set of parameters comprising of output variance and lengthscale for the Matern kernel. From the learnt set of parameters, we take a conservative model by

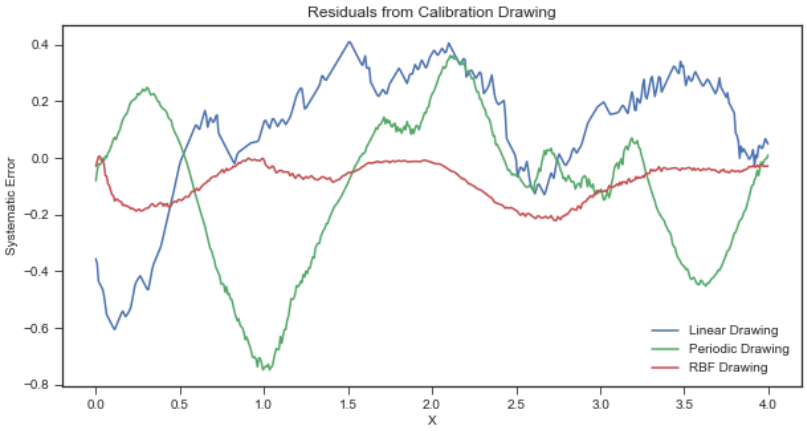

Figure 3.4: Residuals of calibration drawing.

choosing the highest output variance and smallest lengthscale. The value of these parameters is fixed in the proposed model under the assumption that the error made by the user during drawing remains similar.

3.3 Proposed Model

The proposed model places a GP prior on the underlying function for observation data with a Gaussian noise model. For expert data, we take an additive GP model for the underlying function and the error in drawing along with Gaussian noise.

$$f_{obs}(x_{obs}) \sim \mathcal{GP}_{obs}(0\,,\, k_{obs}(x, x')),$$
$$f_{exp}(x_{exp}) \sim \mathcal{GP}_{exp}(0\,,\, k_{exp}(x, x')),$$
$$f_{error}(x_{error}) \sim \mathcal{GP}_{error}(0\,,\, k_{error}(x, x')),$$
$$y_{obs} = f_{obs} + \epsilon_{obs},$$
$$y_{exp} = f_{exp} + f_{error} + \epsilon_{exp}.$$

The hyperparameters of the expert GP ($\mathcal{GP}_{exp}$) and observed GP ($\mathcal{GP}_{obs}$) are shared and jointly optimised. The error model for drawing is fixed ($\mathcal{GP}_{error}$) after calibration and ϵ_{exp} and ϵ_{obs} are the respective expert noise and observation noise.

3.4 Kernel Selection Criterion

To select the right kernel, we needed a kernel selection criteria. The marginal likelihood, as discussed earlier, is the most appropriate because it balances the data fit and complexity of the model. However, calculating marginal likelihood integral is intractable as it requires integrating over the parameter space of kernel. We can approximate this intractable integral by using the Bayesian information criterion (BIC) [18] [23].

$$BIC = k\ln(n) - 2\ln(\hat{L}),$$

where k is the number of parameters estimated by the model, n is the number of data points, and $\hat{L}$ is the marginal likelihood. The other option is to select the kernel based on the predictive capacity of the model. The cross-validation score gives the predictive capacity of the model. However, calculating the cross-validation score for the model can be computationally expensive. The cross-validation score is often approximated using Akaike information criterion (AIC) [1], or Pareto smoothed leave-one-out cross validation (PSIS-LOO) [20]. AICc [4] is a variant of AIC suitable for small sample size as AIC can overfit [10] [5] [9].

Selecting the right kernel with small sample size is difficult because various models can explain little data and machine learning methods cannot learn the pattern from the little dataset because there is not enough information. There have been several studies that have compared the different criterion [2] [3] [21]. We chose BIC because we had the true model in the set of candidates while selecting the kernel and AIC tends to overfit for small sample size.

Chapter 4

Experimental Results

This chapter discusses the Experiments and presents the results. In section 4.1, we recover the true kernel using simulated expert data, and in section 4.2, we recover the true kernel from actual expert drawing followed by section 4.3, where we conduct a user study for the usability of the proposed method. Each Experiment is followed by a description of the setup and the results. We also discuss about learning the true lengthscale in section 4.4.

4.1 Experiment 1

This Experiment aims to quantify the benefit of prior elicitation from expert drawing data in an optimistic setting where the expert can draw the true function.

4.1.1 Setup

We test the method in a grid specifying a combination of the following parameters : number of observation data points, number of expert data points, observation residual variance, expert residual variance and value of the hyperparameters from which ground truth is generated. The hyperparameters are the lengthscale (L) for RBF kernel, lengthscale (L) and periodicity (P) for periodic kernel and intercept (I) for the linear kernel. The parameter grid is shown in Table 4.1.

	No. of Exp.	**No. of Obs.**	**Noise (obs)**	**Noise (exp)**
High	256	256	1	1
Low	16	16	0.2	0.2

Table 4.1: Parameter grid.

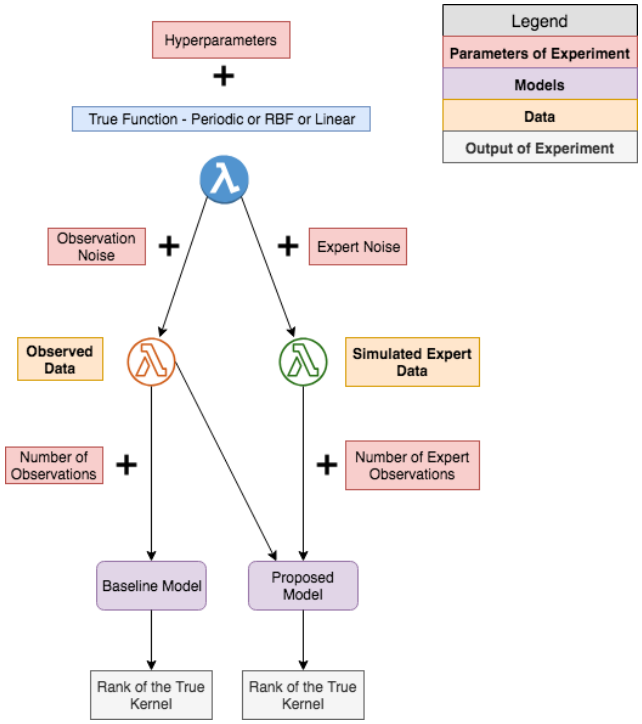

Figure 4.1: Experiment 1 setup.

The ground truth is generated from RBF, periodic or linear kernel. We add some Gaussian noise to simulate observed data and expert data. As discussed previously, the baseline model gets only observed data whereas the proposed model gets expert data along with the observed data. Both models are restricted to search in between the three kernels: RBF, periodic and linear. We selected the best model according to the lowest BIC values for the baseline model and proposed model respectively. We then calculate the average rank of the true kernel with confidence intervals from the result of over 20 iterations.

To calculate BIC, we need to find the maximum likelihood estimate (MLE). The estimate is found using a conjugate gradient optimisation technique. The optimisation can get stuck in bad local maxima, so we use 10 random restarts and then take the maximum value of log likelihood of the model from these estimates.

4.1.2 Results and Discussion

The results for Experiment 1 are shown in Table 4.2, Table 4.3 and Table 4.4 for linear, RBF and periodic kernel respectively. The results show that the

No. of Obs	Obs noise	Hyper parameters (I)	Expert noise	Baseline Model Rank	Proposed Model Rank
low	low	low	low	2.25 (0.54)	**1.10** (0.44)
			high	2.45 (0.59)	**1.55** (0.74)
		high	low	1.45 (0.67)	**1.05** (0.22)
			high	1.35 (0.65)	**1.20** (0.60)
	high	low	low	2.50 (0.67)	**1.15** (0.36)
			high	2.55 (0.59)	**1.40** (0.58)
		high	low	2.15 (0.91)	**1.00** (0.00)
			high	2.15 (0.85)	**1.00** (0.00)
high	low	low	low	1.55 (0.59)	**1.00** (0.00)
			high	1.80 (0.75)	**1.15** (0.36)
		high	low	1.05 (0.22)	**1.00** (0.00)
			high	**1.00** (0.00)	**1.00** (0.00)
	high	low	low	2.35 (0.48)	**1.10** (0.30)
			high	2.00 (0.63)	**1.50** (0.81)
		high	low	1.25 (0.54)	**1.10** (0.44)
			high	1.35 (0.65)	**1.10** (0.44)

Table 4.2: Average rank for linear kernel.

No. of Obs	Obs noise	Hyper parameters (L)	Expert noise	Baseline Model Rank	Proposed Model Rank
low	low	low	low	1.80 (0.68)	**1.00** (0.00)
			high	1.45 (0.59)	**1.05** (0.22)
		high	low	1.30 (0.46)	**1.10** (0.30)
			high	1.60 (0.49)	**1.20** (0.40)
	high	low	low	2.35 (0.73)	**1.00** (0.00)
			high	2.05 (0.59)	**1.00** (0.00)
		high	low	1.90 (0.70)	**1.05** (0.22)
			high	1.75 (0.54)	**1.00** (0.00)
high	low	low	low	1.10 (0.30)	**1.00** (0.00)
			high	1.15 (0.36)	**1.05** (0.22)
		high	low	1.10 (0.30)	**1.00** (0.00)
			high	1.15 (0.36)	**1.00** (0.00)
	high	low	low	1.00 (0.00)	1.05 (0.22)
			high	1.10 (0.30)	**1.05** (0.22)
		high	low	1.05 (0.22)	**1.00** (0.00)
			high	1.15 (0.36)	**1.10** (0.30)

Table 4.3: Average rank for RBF kernel.

No. of Obs	Obs noise	Hyper parameters (L & P)	Expert noise	Baseline Model Rank	Proposed Model Rank
low	low	low	low	1.15 (0.36)	**1.05** (0.22)
			high	1.45 (0.59)	**1.15** (0.36)
		high	low	1.15 (0.36)	**1.10** (0.30)
			high	1.35 (0.48)	**1.25** (0.43)
	high	low	low	1.20 (0.40)	**1.05** (0.22)
			high	1.60 (0.58)	**1.25** (0.54)
		high	low	1.25 (0.43)	**1.05** (0.22)
			high	1.40 (0.58)	**1.35** (0.48)
high	low	low	low	1.00 (0.00)	1.05 (0.22)
			high	1.00 (0.00)	1.05 (0.22)
		high	low	1.10 (0.30)	**1.05** (0.22)
			high	1.00 (0.00)	1.10 (0.30)
	high	low	low	**1.00** (0.00)	**1.00** (0.00)
			high	1.20 (0.40)	**1.15** (0.36)
		high	low	1.35 (0.48)	**1.00** (0.00)
			high	**1.25** (0.43)	**1.25** (0.54)

Table 4.4: Average rank for periodic kernel.

proposed model average rank of true kernel is better than baseline model in most of the cases. The better average rank of true kernel can be attributed to the effect of more data. When the number of observations are high in Table 4.3 and Table 4.4, we can see that the proposed model is choosing the true kernel fewer times than baseline model. The log likelihood term in BIC for more complex model can dominate the penalty term which becomes constant with increase in observations. This has also been seen in comparison of AIC and BIC by Vrieze [21].

4.2 Experiment 2

This Experiment aims to validate our results on synthetic data in a real-world setting where the expert can draw the true function approximately.

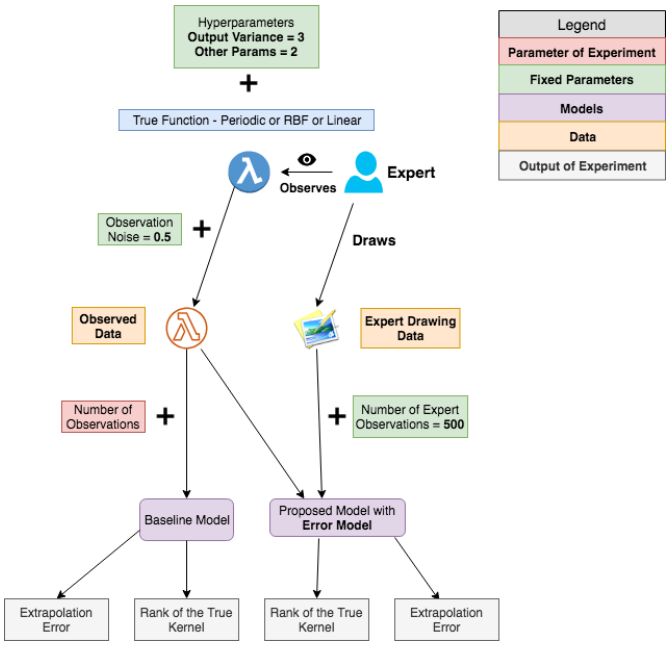

Figure 4.2: Experiment 2 setup.

4.2.1 Setup

This Experiment is the practical version of Experiment 1 with varying number of observation points and every other parameter fixed. A human user is drawing the true function as shown to him. As described earlier, we use MLE with 10 random restarts.

We generate the ground truth from RBF, periodic or linear kernel with output variance set as 3 and other parameters of the kernel set as 2. The data is divided into training set where the value of X lies from 0 to 4 with 400 points and 4-5 with 100 points as test set .The expert can see only the training set part of the true function and draws on scale with the data. To generate the observation data, we add a fixed Gaussian noise of 0.5 to the training dataset and vary the number of observation points. The number of data points taken from expert drawing is fixed at 500 points. We run the simulation 20 times for each number of observation points. We calculate the average rank with confidence interval along with the extrapolation error in terms of RMSE on the test set.

4.2.2 Calibration

For learning the error model of the user, we ask the expert to draw three known functions generated from linear, periodic and RBF kernel respectively. We then calculate the residuals for each of the three drawings and then learn the lengthscale and output variance of noise for each of the three drawings. Finally, we take a conservative model by selecting maximum output variance and smallest lengthscale so that we can capture the irregularities in the drawing. We fix the value of hyperparameters in the error model.

4.2.3 Results and Discussion

For the linear kernel, Table 4.5 shows that the baseline model recovers the true kernel better than the proposed model resulting in a better extrapolation performance. For the RBF kernel, the proposed model performs significantly better than the baseline model with little data as seen in Table 4.6. As data increases, the performance of the baseline model begins to improve. For the periodic kernel, the proposed model outperforms the baseline model in recovering the true kernel leading to better extrapolation performance as seen in Table 4.7.

Periodic kernel and RBF kernel with high hyperparameter values, for example, lengthscale can model the data from the linear kernel. Whenever the expert draws, the line is usually not straight, and hence, the proposed model tends to select the RBF kernel sometimes to model the data generated from linear kernel. We can see in Table 4.7 that as the number of observations increase the average rank for periodic kernel doesn't improve for the baseline model which is not the case for RBF kernel in Table 4.6. This happens because BIC tends to select the RBF kernel having less number of parameters than periodic kernel which can also model the data from generated from

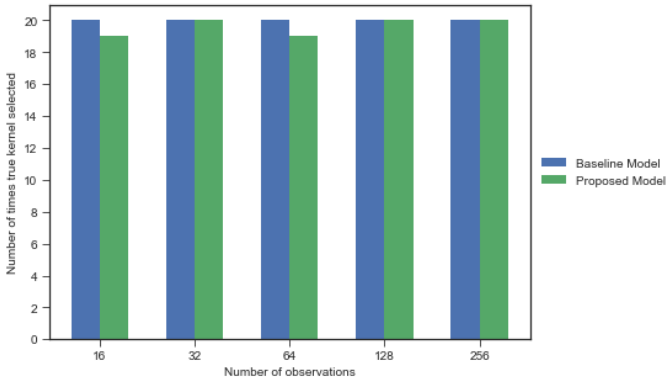

Figure 4.3: Number of times true kernel was discovered (linear).

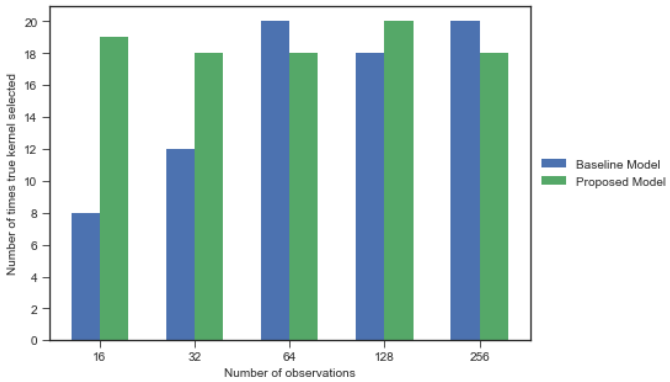

Figure 4.4: Number of times true kernel was discovered (RBF).

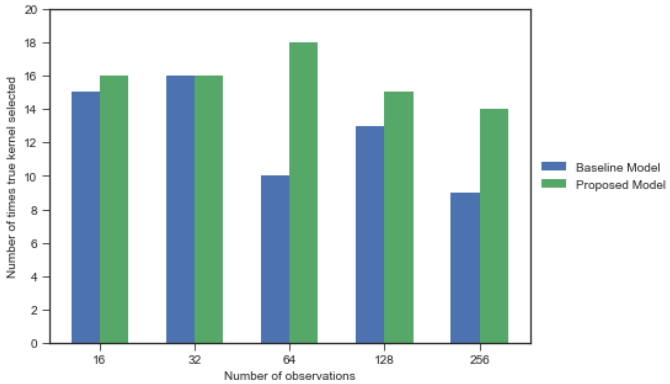

Figure 4.5: Number of times true kernel was discovered (periodic).

No. of Obs	Average Rank		Extrapolation Error	
	Baseline Model	Proposed Model	Baseline Model	Proposed Model
16	1.00 (0.00)	1.05 (0.22)	0.27 (0.19)	0.35 (0.43)
32	**1.00** (0.00)	**1.00** (0.00)	0.16 (0.13)	0.19 (0.14)
64	1.00 (0.00)	1.05 (0.22)	0.11 (0.07)	0.12 (0.09)
128	**1.00** (0.00)	**1.00** (0.00)	**0.08** (0.05)	**0.08** (0.06)
256	**1.00** (0.00)	**1.00** (0.00)	0.07 (0.06)	0.08 (0.06)

Table 4.5: Average rank and extrapolation error for linear kernel.

No. of Obs	Average Rank		Extrapolation Error	
	Baseline Model	Proposed Model	Baseline Model	Proposed Model
16	1.60 (0.49)	**1.05** (0.22)	1.27 (0.47)	**0.61** (0.51)
32	1.40 (0.49)	**1.10** (0.30)	1.02 (0.71)	**0.64** (0.51)
64	1.00 (0.00)	1.10 (0.30)	0.42 (0.28)	0.46 (0.36)
128	1.10 (0.30)	**1.00** (0.00)	0.59 (0.43)	**0.52** (0.30)
256	1.00 (0.00)	1.10 (0.30)	0.30 (0.20)	0.39 (0.38)

Table 4.6: Average rank and extrapolation error for RBF kernel.

No. of Obs	Average Rank		Extrapolation Error	
	Baseline Model	Proposed Model	Baseline Model	Proposed Model
16	1.25 (0.43)	**1.20** (0.40)	0.77 (0.53)	**0.70** (0.64)
32	**1.20** (0.40)	**1.20** (0.40)	0.55 (0.60)	0.58 (0.63)
64	1.50 (0.50)	**1.10** (0.30)	0.95 (0.73)	**0.44** (0.46)
128	1.35 (0.48)	**1.25** (0.43)	0.65 (0.70)	**0.52** (0.66)
256	1.55 (0.50)	**1.30** (0.46)	0.89 (0.70)	**0.55** (0.66)

Table 4.7: Average rank and extrapolation error for periodic kernel.

periodic kernel. The proposed model can distinguish this difference due to extra data from expert.

To select the right kernel, we need to determine the dependency on expert data. The shape of the expert drawing is determined by which points are selected and how much we want to depend on expert drawing depends on the number of points taken from an expert drawing. Hence, the number of expert points can also be treated as a parameter which introduces the new problem of preserving the right shape. We eliminate this problem by using the error model. The error model captures the error in the drawing which makes it easier to separate the signal rather than optimising over the number of points.

In Figure 4.6, the observation GP is trained on observed data shown in blue points whereas the expert GP is trained on expert drawing data shown in blue points. The expert GP is an additive GP made of the signal GP and the error GP. We can see a slight bump in error GP between 0 and 0.5 which models the deviation in expert drawing points from the true function shown in blue.

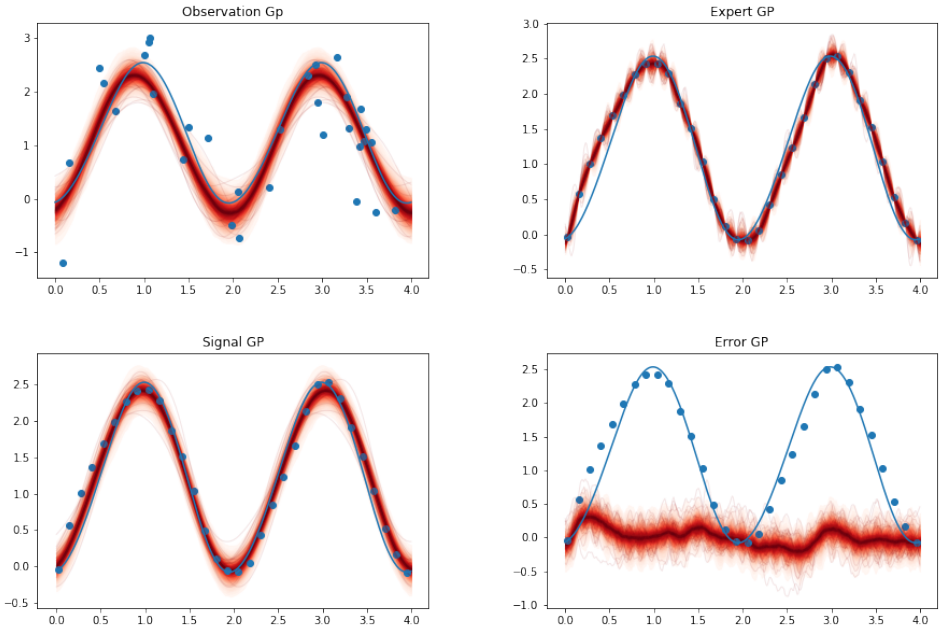

Figure 4.6: Working of the Error Model.

For the Experiment, the latent functions to be drawn were generated with high output/signal variance and high hyperparameter values such as lengthscale and period. The reason for choosing high mean magnitude was so that the signal does not get drowned in the noise. Any large deviation

made while drawing the function by the expert could lead to the selection of an error model with high magnitude. This would lead to selection of wrong kernel because the error GP can capture some parts of the signal GP. The second reason for choosing high hyperparameters values was so that the functions are smooth and can be easily drawn.

4.3 Experiment 3: User Study

We conducted a proof of concept user study to find out whether users can improve the performance of the model with the help of drawing. As we do not have any expert, we simulate the expert by showing naive users what they had to draw. The user study was conducted on 5 university students.

4.3.1 Setup

The user study is divided into four phases where each phase consists of three tasks. The first phase is where the user asked to practice drawing on the interface. Figure 4.7 shows the first phase for linear kernel. The three tasks are to draw three types of functions drawn from the linear kernel, RBF kernel and periodic kernel. During this phase, the users are exactly shown what they have to draw. In the background, we use the training phase for calibration and learning the error model for the user.

The second phase is the naive phase where users are shown a set of observations having some error and asked to draw the function that could have generated those points. Figure 4.8 shows the second phase for linear kernel. Again, the three tasks consist of drawing the unknown function for the observed data (different from that in the first phase) generated from the linear kernel, RBF kernel and periodic kernel. The third phase is the partial knowledge phase where the user is shown the type of function that has generated the data and is also shown some clue points (along with observed data) through which the drawn function must pass. Figure 4.9 shows the third phase for linear kernel. In the fourth phase, we show the specific function which needs to be drawn by the user. Figure 4.10 shows the fourth phase for linear kernel.

The first assumption is that the proposed model would perform better than the baseline model. The second assumption is that as the users would get more information with each phase, the performance of the proposed model would increase due to better drawing.

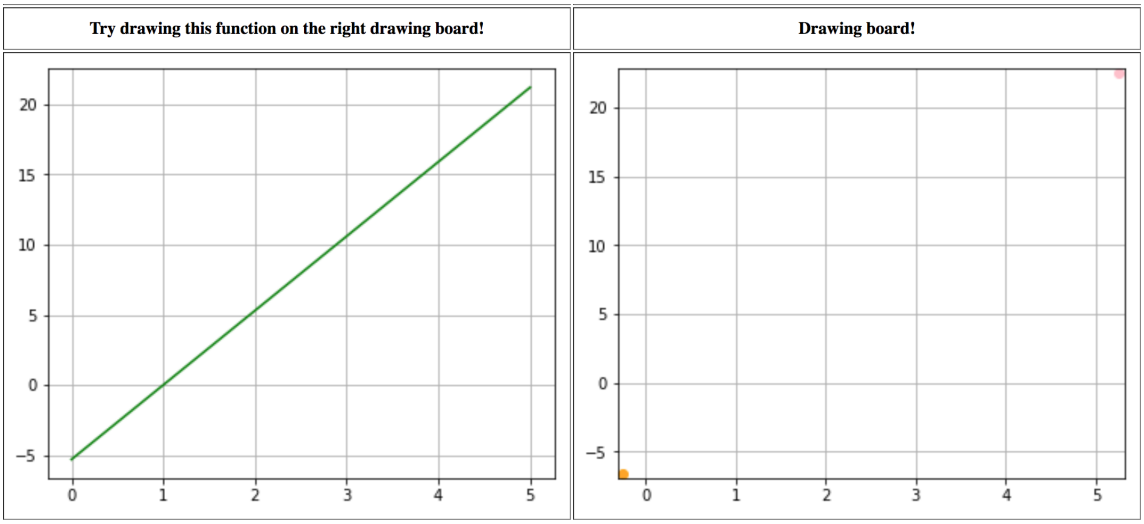

Figure 4.7: Phase 1: Practice drawing for linear kernel. This data is different from the data tested in phase 1, 2 and 3.

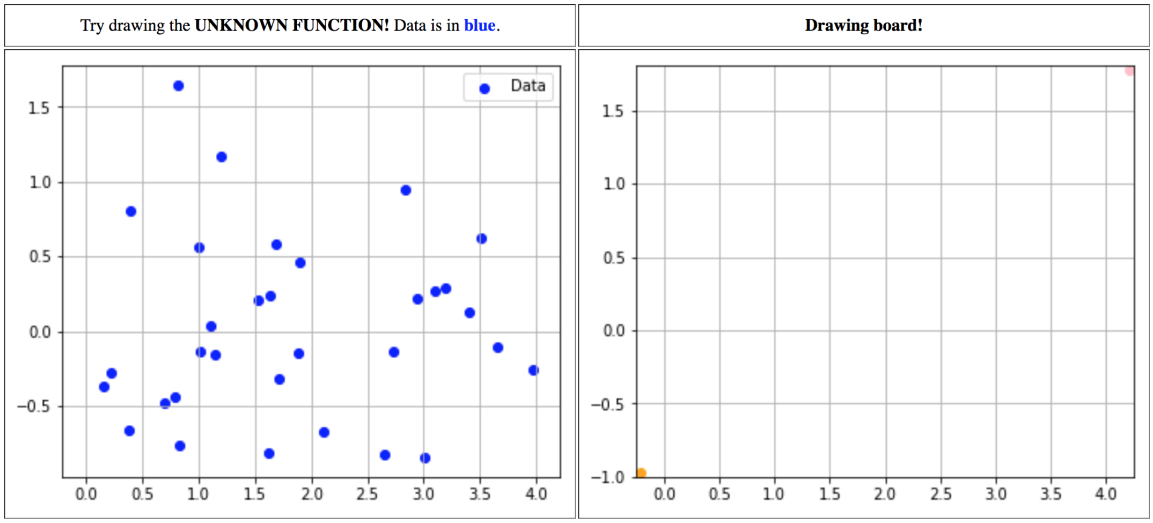

Figure 4.8: Phase 2: Unknown function for data from linear kernel.

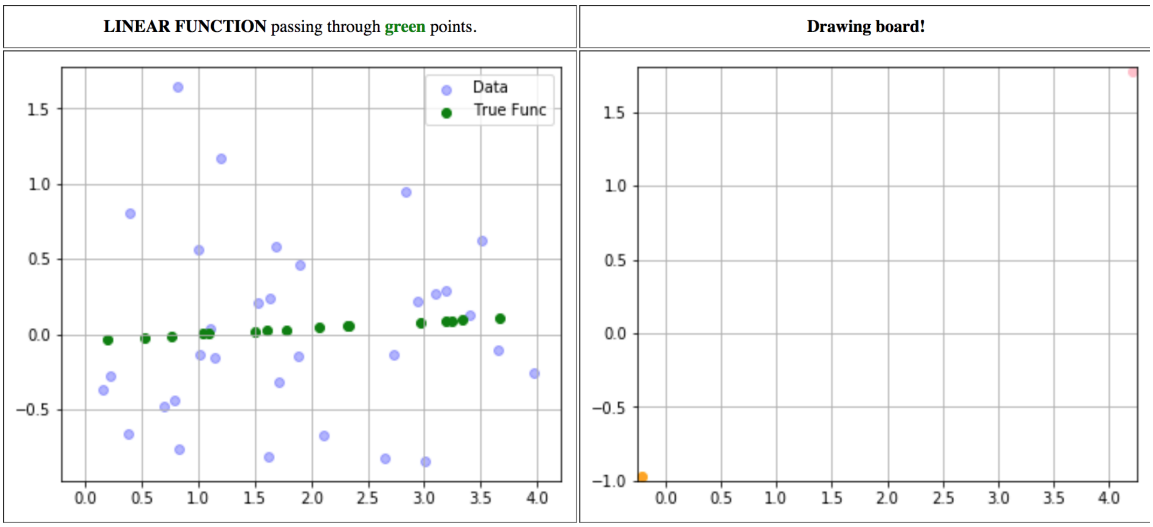

Figure 4.9: Phase 3: Hints for unknown function for data from linear kernel. The y axis looks different than that from Figure 3 because of observation data shown in blue.

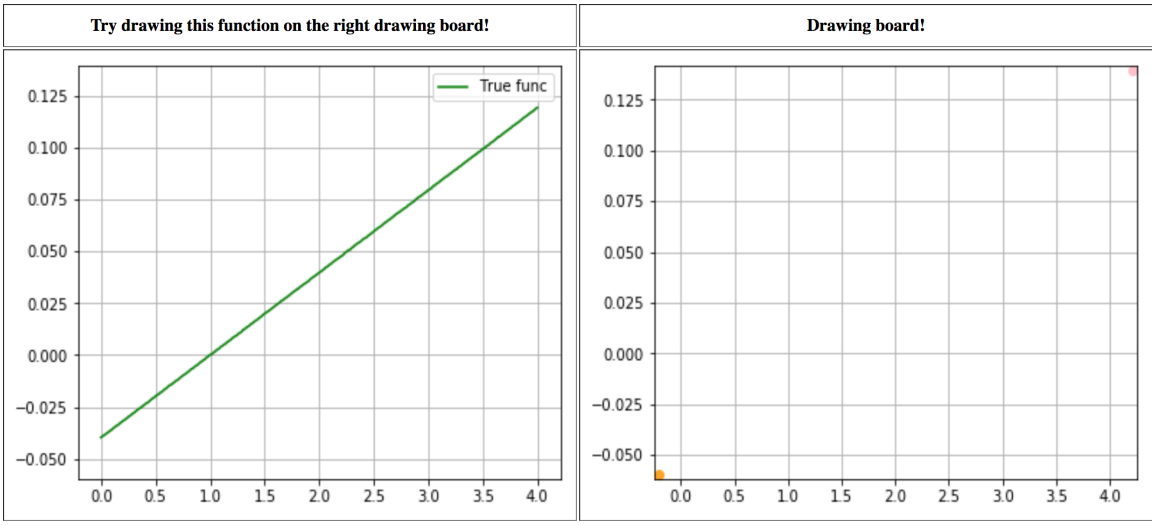

Figure 4.10: Phase 4: Actual function from which data is generated.

4.3.2 Results and Discussion

The result of the simulations on drawings from user study are in the Appendix due to verbosity. We measure the error between user drawing and original functions using RMSE. Table 4.9 shows the RMSE of the users during the practice phase and Table 4.8 shows the RMSE of the users in each drawing. Table 4.9 shows that all the users made maximum mistake in drawing Task 1 where the function was linear. The assumption that the user would not make more error in drawing than in the practice phase did not hold for our user study. This means that the fixed conservative error model, learnt during the practice phase, cannot account for the variation in error during drawings because sometimes the maximum error made by the user while drawing can be larger than the error in the practice phase.

Phase	Task	User 1	User 2	User 3	User 4	User 5	Task Avg
Naive Phase	Task 1	0.379	**0.253**	0.539	0.282	0.392	0.369
	Task 2	0.715	0.653	0.349	**0.187**	0.854	0.552
	Task 3	0.258	0.259	0.406	**0.198**	0.322	0.288
Partial Phase	Task 4	0.031	0.027	0.066	0.033	**0.017**	0.035
	Task 5	0.371	0.335	0.272	**0.096**	0.280	0.271
	Task 6	0.506	**0.135**	1.074	0.176	0.246	0.427
Expert Phase	Task 7	0.006	0.009	0.016	**0.003**	0.008	0.008
	Task 8	0.349	0.154	0.209	**0.102**	0.169	0.197
	Task 9	0.265	0.215	0.510	**0.104**	0.429	0.305
User Average Error		0.320	0.227	0.382	**0.131**	0.302	

Table 4.8: RMSE error of users in drawing tasks.

Phase	Task	User 1	User 2	User 3	User 4	User 5
Practice Phase	Task 1	**0.894**	**1.250**	**1.141**	**0.611**	**1.720**
	Task 2	0.100	0.135	0.139	0.071	0.086
	Task 3	0.104	0.072	0.097	0.041	0.364
Error Model Magnitude		0.705	0.909	0.843	0.498	1.233

Table 4.9: RMSE error of users in practice drawing tasks.

For a linear kernel that is Task 1, 4 and 7 the error decreases drastically in the partial phase when the user is shown a few points through which function passes and provided hint that the function is linear. For periodic kernel that is Task 2, 5 and 8 the error decreases in each phase, but it is still difficult

for users to get the periodicity right. For RBF kernel that is Task 3, 6 and 9 the error increases in partial phase and then decreases in the expert phase. This shows that it isn't necessary that the extra information would increase the quality of expert data because the result also depends on how the expert draws each time. However, drawing such conclusions from small number of users is not right.

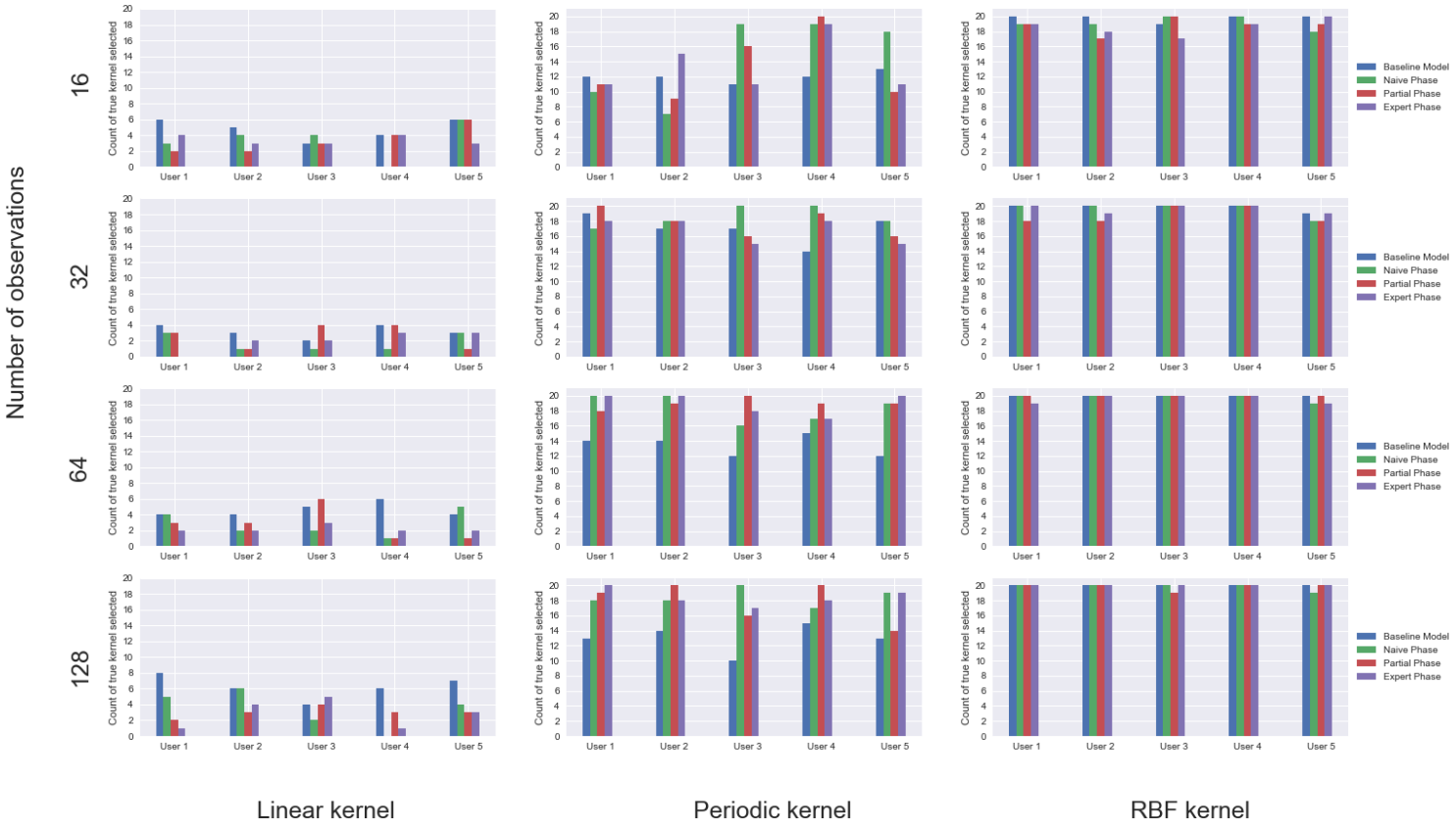

Figure 4.11: This Figure compares number of times the true kernel was selected for each user for different number of observation across different phases. The x axis of the grid denotes the kernel and the y axis of the grid shows number of observations.

For tasks 1, 4 and 7 associated with linear kernel, we can see that all the models performed poorly by consistently selecting the RBF kernel. The drawing data did not help further in kernel selection across all users. The

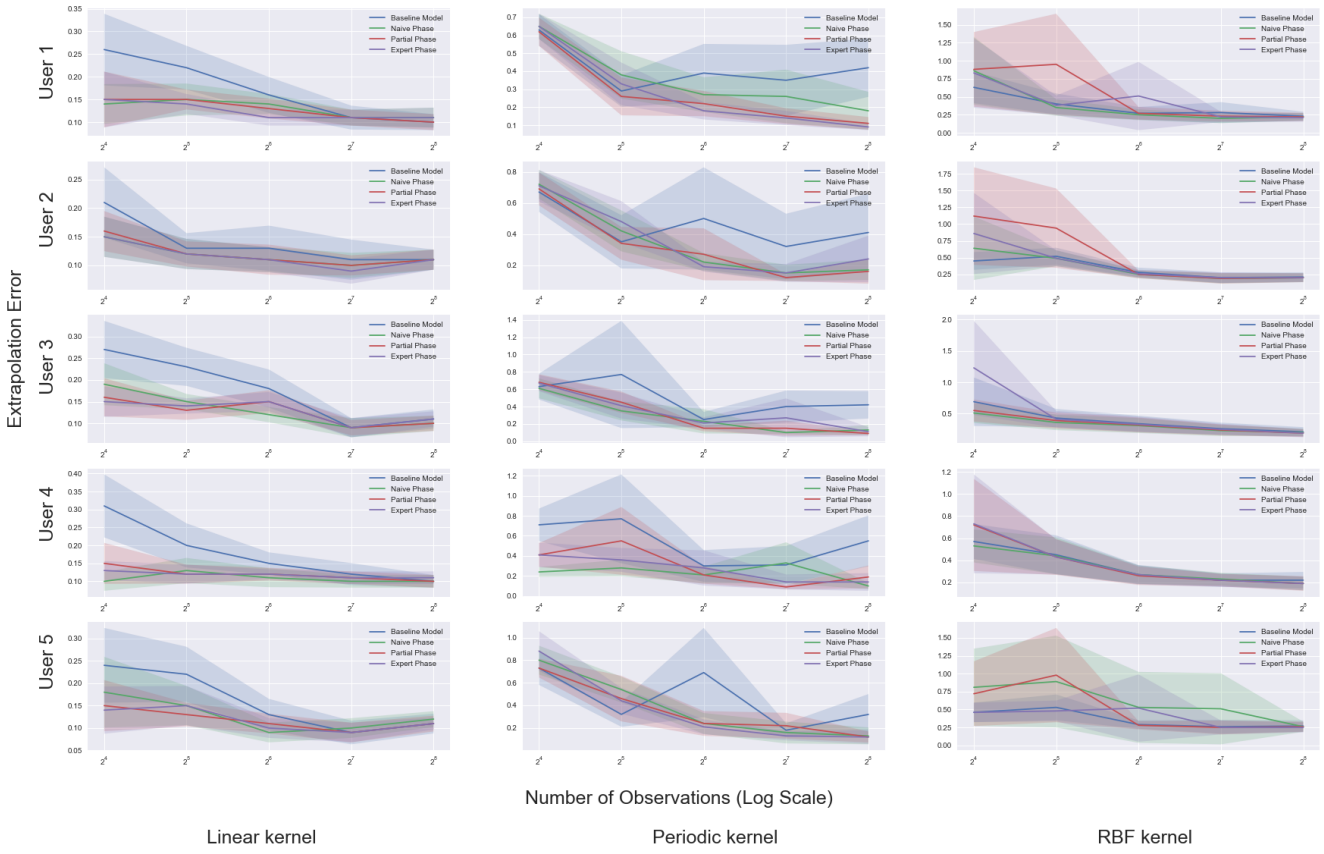

Figure 4.12: This Figure compares extrapolation error for each user for different number of observation across different phases and kernels. The x axis of each subplot is number of observations (log scale) and the y axis shows extrapolation error.

selection of RBF kernel instead of the linear kernel can be attributed to BIC because even the baseline model which doesn't use expert data did not select linear kernel with an increasing number of observations. However, this results contradicts with the good performance shown in Experiment 2. In order to further analyze this issue, we plot the BIC values for the three kernels with increasing number of observations in user study and Experiment 2. The result of this analysis are obtained from 10 simulations of the data and only the mean is plotted for clarity. Figure 4.13 shows that the BIC values of RBF kernel is lower than linear kernel in user study leading to selection of the RBF kernel. Figure 4.14 shows that BIC values of linear kernel is lower than RBF kernel in Experiment 2 leading to selection of the linear kernel. Linear kernel and RBF kernel have the same number of parameters which means they have the same penalty term. RBF kernel can explain data generated from linear kernel with a high lengthscale. Hence, BIC is responsible for the poor performance for selection of linear kernel in user study.

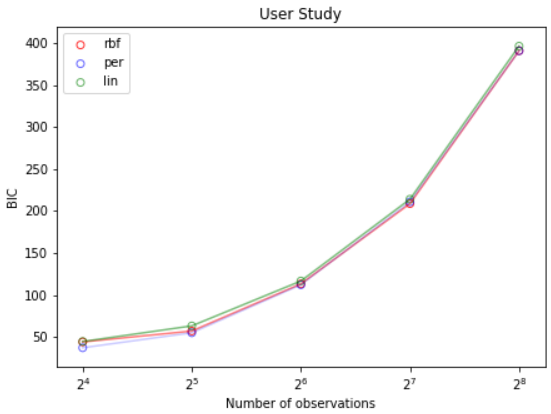

Figure 4.13: This Figure shows mean BIC value of baseline model vs the number of observations for kernel selection (linear) in user study.

For tasks 2, 5 and 8 associated with Periodic kernel, the proposed model performs better than the baseline model. For tasks 3, 6 and 9 associated with RBF kernel, all models perform equally well. The RBF function was relatively easy to draw for most of the users. User 4 has obtained excellent results which can be attributed to his precise drawing.

Extrapolation performance should increase with the number of observations which means the RMSE should decrease on the test set. Figure 4.12 shows that the extrapolation performance of the baseline model decreases sometimes with the number of observations. This can be explained on the

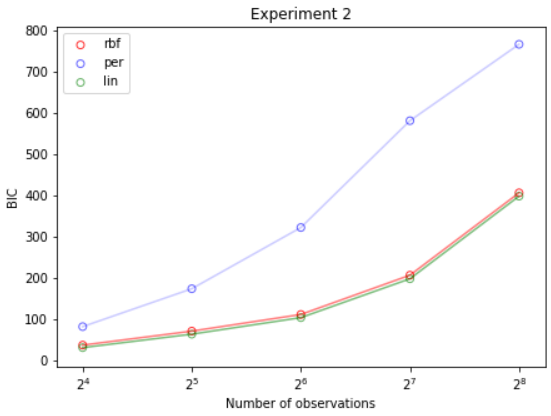

Figure 4.14: This Figure shows mean BIC value of baseline model vs the number of observations for kernel selection (linear) in Experiment 2.

basis of kernel selection. If increasing the number of observations is leading to selection of wrong kernel, extrapolation performance is likely to decrease. This raises the question of why wrong kernel is selected with increasing number of observations. This happens because log penalty term in BIC does not increase with the same rate as the data likelihood term on increasing the number of observations leading to selection of wrong kernel. Figure 4.12 also shows that the proposed model performs better than the baseline model in low data region and as the number of observations increase both model perform at par. Especially for the periodic kernel with increasing number of observations, the expert data helps in selection of true kernel.

Summarizing the results of user study, the proposed model definitely improves extrapolation performance compares to baseline model for all users in multiple settings validating our first assumption. For kernel selection task, the poor performance for linear kernel can be attributed to BIC. For RBF and periodic kernel, Figure 4.11 shows that drawing data helped in selecting the right kernel for most users. As the cohort for our user study was small, we cannot comment much about users. Still, almost all users achieved good performance irrespective of how good they were at drawing. However, our second assumption of improved performance in consecutive phases does not hold because more information regarding the function to be drawn does not necessarily decrease the drawing error which is presented in task avg column of Table 4.8.

4.4 Learning the Lengthscale

We also performed preliminary Experiments to identify the lengthscale of an RBF kernel with the help of simulated expert data. However, this turned out to be challenging. In spite of setting informative log-normal priors and half-normal priors, we were not able to learn the true lengthscale, both in the proposed model and the baseline model.

To realise the benefit of learning the true lengthscale, we fixed the value of lengthscale to the true value in a proxy model and compared the performance to the baseline model on a test set using root mean square error (RMSE). The baseline model still performed better compared to the proposed model because of bias-variance trade-off. The proposed model is a more complex model compared to the baseline model and has higher variance in its predictions. The proposed model might have a lower bias, but the variance in its predictions can lead to higher average RMSE over 20 simulations.

From the simulations, we realised that the GP can find the optimal hyperparameters but not the actual hyperparameter value because there is weak identifiability in kernel hyperparameters for GPs as they are a flexible class of model [15]. Kernels can be represented in the frequency domain. The power spectrum of the RBF kernel is given by

$$S(s) \propto \alpha^2 \ell \exp(-2\pi^2 \ell^2 s^2).$$

The above expression is interpreted as the weight the kernel gives to eigenfunction with frequency s. We can see that the output variance α^2 term and lengthscale ℓ term occur together. Increasing output variance (α^2) increases power to all frequencies whereas increasing lengthscale(ℓ) increases power to low frequency signal at the expense of the high-frequency signal. For example, we can achieve the same effect as that of high lengthscale function by decreasing the output variance and keeping the lengthscale low as seen in Figure 4.15. In case of little data, we don't have enough resolution to distinguish between the increase in lengthscale or the decrease in output variance.

The hyperparameters need to be constrained by careful prior formulation and data to learn the true hyperparameters. Little data alone is not informative enough to learn the true hyperparameters. Careful prior formulation means that the expert should think in terms of the number of parameters in the model when setting the prior for lengthscale. However, we do not expect the expert in our method to have any knowledge regarding Bayesian statistics, so we decided to select the kernel with the help of expert data.

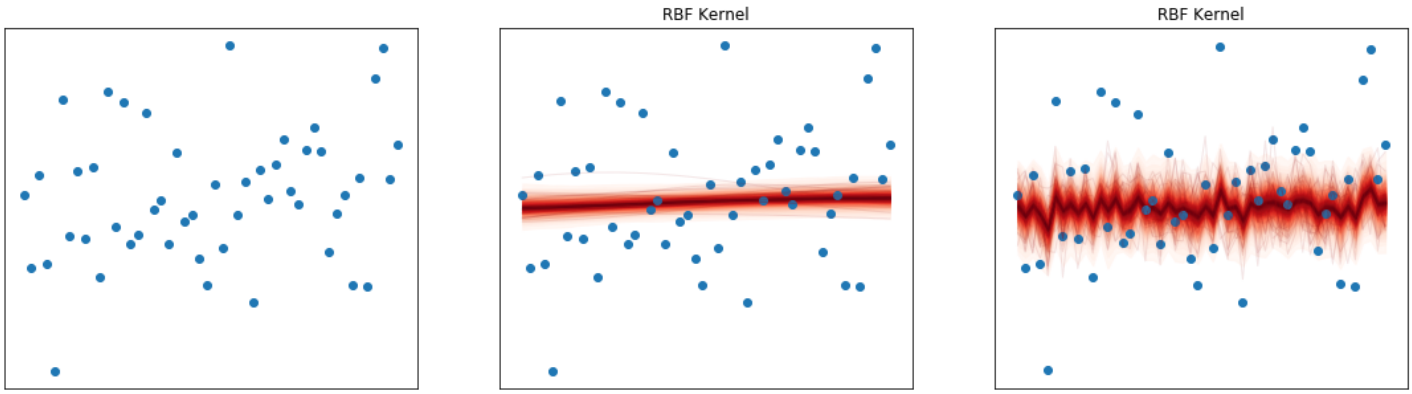

Figure 4.15: On the left side, we see the observed data. The middle figure shows a fit with high lengthscale function to the observed data whereas the right figure shows a fit with small lengthscale function to the observed data. In the right, we decrease the output variance (magnitude of mean) to achieve the same effect as a high lengthscale function.

Chapter 5

Discussion

This chapter summarises the work done in the thesis and presents some new directions for future work.

5.1 Conclusion

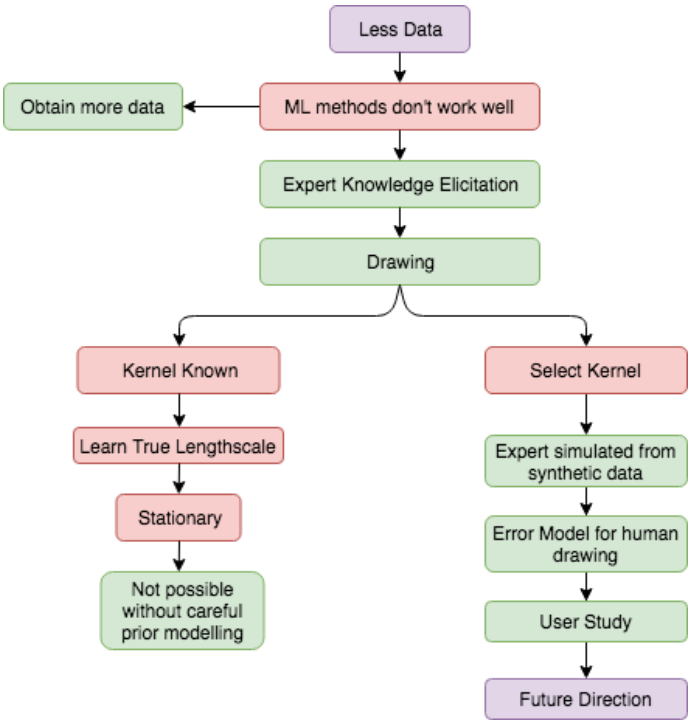

Figure 5.1: Conclusion of thesis.

In the first chapter, we explained the problem that most machine learning methods do not work well with little data. Unlike humans who have context about other factors affecting the problem, machine learning only depends on the dataset. We have two choices, either to obtain more data or to take the help of an expert. The expert can help us by providing more data or in selecting the right model. In this thesis, we have discussed the option of eliciting knowledge from an expert using drawing. The expert tells us by drawing how the function should fit and doesn't need to perform any complex modelling tasks. Combining the drawing data and observations, we explored different ways to improve the performance of the model compared to the model using only observations.

We decided to select the kernel with the help of elicited drawing data. For kernel selection, we chose the Bayesian Information Criterion (BIC) as the true model was present in the selection procedure. Firstly, we simulated the expert using synthetic data. The proposed model outperformed the baseline model in nearly all cases by having a better average rank for the true kernel. The increase in performance can be attributed merely to the selection of the right kernel from the increase in data. Secondly, we performed an experiment where the expert was simulated by showing the user what he has to draw. In this experiment, we learnt that the magnitude of the signal needs to be higher magnitude than the magnitude of the error model. When this is not the case, the error model can capture some parts of the signal leading to the selection of the wrong kernel. After getting positive results in both experiments, we went forward with user study to test the usability of the system.

The user study showed that more information regarding the unknown function did not necessarily lead to more accurate elicitation of knowledge. The elicited knowledge also depends on how the user processes that information and translates it into precise drawing movements. User 4 who was more careful at drawing from all other users was able to incorporate this information and got better results than other users. Capturing the ideal error for each drawing drawn by expert is not possible in the case of the unknown function. Few solutions to capture the error in the human drawing are proposed below. We can ask the expert to draw the unknown function multiple times and take the average of drawings as the correct drawing. The more proper methodology would be to set up a hierarchal model over the multiple drawing data from the expert for the same unknown function. Considering all of this, the proposed model still performed better compared to the baseline model for periodic kernel and RBF kernel in kernel selection task in user study. The extrapolation error was less for the proposed model compared to the baseline model for multiple settings in the user study.

We couldn't compare the proposed model to baseline model in lower data

regions because of the limitations of BIC. For BIC, we need $n \gg k$ where n is the number of observation and k is the number of parameters in the model. It would be interesting to perform the experiment with another criterion such as AICc and analyze the results. Particularly in linear kernel selection task on Experiment 2, both of the models perform very well whereas both of them perform poorly in user study. The data generated from linear kernel can be explained by RBF kernel and both have the same number of parameters so both have very close values as seen in Figure 4.13 and Figure 4.14. The choice of observed dataset can easily influence the result for linear kernel selection task like in Experiment 2 and user study. For RBF kernel selection task, the linear kernel cannot explain the curved data and the periodic kernel has higher number of parameters, so both baseline model and proposed model perform well in Experiment 2 and user study. For periodic kernel selection task, the baseline model selects RBF kernel with increasing observations as BIC has larger penalty term for more complex models. We had chosen a function with high lengthscale and high periodicity so that it can be easily drawn. Such functions can be explained by the RBF kernel. However, the proposed model selects the periodic kernel because the extra drawing data is not so smooth leading to a higher likelihood term for periodic kernel.

For any expert knowledge elicitation research, eliciting the expert's belief accurately and the correctness of extracted knowledge are two different things. Human drawing has varying systematic error known as drift which we were able to correct partially for using the error model. By incorporating the fixed error model in the proposed model and assuming that the expert draws carefully, we were able to select the true kernel from the set of given models. We assume that the error made by the user remains same throughout the experiment. This assumption is not valid. Hence, like other elicitation methods, we depend on the quality of elicited expert knowledge. The quality of expert knowledge depends on how precisely we can capture a user's drawing error for that particular task.

Apart from the above limitations, drawing can be a more intuitive way to elicit knowledge from experts because we can visually spot patterns from data which gives us insight about the nature of the function. Drawing is an effective way to communicate this knowledge rather than being asked for point estimates for function values at unknown points. Active elicitation methods can also ask for such point estimates or to select from a group of competing models, however, steering the system to what we want can be difficult. Drawing also doesn't require experts to deeply dive into Bayesian statistics to do the modelling needed for their research problem.

We also tried to learn the true lengthscale with help of the expert drawing data and observations. True lengthscale is the value used for generating the

synthetic data of the experiment. The idea was to learn the true lengthscale for stationary kernel and then extend it to non-stationary kernels. We found that learning the true lengthscale was not possible without careful prior formulation. Without a proper prior, the parameter space is not constrained enough by data to determine the true lengthscale. Precise prior formulation implies that the expert can make complex modelling decisions which would invalidate our assumptions for using drawing.

5.2 Future Work

In this thesis, we have restricted ourselves to modelling simple functions. It was difficult to find practical use cases where the expert would know what to draw. Most of the tasks for our naive users, with whom we simulate the expert, could simply be considered as connecting the dots. We need to reconsider our approach to apply this method to practical use cases. To extend this method to datasets, the expert needs to draw complex function accurately. Drawing complex functions accurately can be especially difficult because of the frequent change in hand movements. It might also be difficult to get seasonal variations right for example in time series data. In such cases, it can be beneficial if the expert can decompose what he wants to draw into simpler drawings. These simpler drawings can be combined to form a more complex drawing.

5.2.1 Extrapolation

As already discussed earlier, Gaussian process falls back to it's mean away from the training data points depending on the mean and kernel. One way of solving this problem is to have a more complex mean function in case we want to have certain behaviour away from training points. In expert knowledge elicitation by drawing, we can ask the expert to extrapolate into the future. We can use the learned model to select the best data points from this additional task and include them in our observation dataset for anchoring the Gaussian process.

5.2.2 Generalized Additive Model

Generalized additive model is a non linear model in which the unknown response variable depends on a linear combination of non linear predictor variables. We can extend this method to Generalized additive model where the expert can draw for each predictor variable. After selecting the kernel

for each predictive model, we can combine the respective Gaussian processes to get final prediction based on all predictor variables. In case, where the expert does not know about the relationship between response variable and predictor variable, they can choose to skip it.

Bibliography

[1] Akaike, H. Information theory and an extension of the maximum likelihood principle. In *Selected papers of Hirotugu Akaike*. Springer, 1998, pp. 199–213.

[2] Anderson, D., and Burnham, K. Model selection and multi-model inference. *Second. NY: Springer-Verlag* (2004), 63.

[3] Burnham, K. P., and Anderson, D. R. Multimodel inference: understanding AIC and BIC in model selection. *Sociological methods & research 33*, 2 (2004), 261–304.

[4] Cavanaugh, J. E. Unifying the derivations for the Akaike and corrected Akaike information criteria. *Statistics & Probability Letters 33*, 2 (1997), 201–208.

[5] Claeskens, G., and Hjort, N. L. Model selection and model averaging. Tech. rep., Cambridge University Press, 2008.

[6] Duvenaud, D. *Automatic model construction with Gaussian processes*. PhD thesis, University of Cambridge, 2014.

[7] Duvenaud, D., Lloyd, J. R., Grosse, R., Tenenbaum, J. B., and Ghahramani, Z. Structure discovery in nonparametric regression through compositional kernel search. *arXiv preprint arXiv:1302.4922* (2013).

[8] Garthwaite, P. H., Kadane, J. B., and O'Hagan, A. Statistical methods for eliciting probability distributions. *Journal of the American Statistical Association 100*, 470 (2005), 680–701.

[9] Giraud, C. *Introduction to high-dimensional statistics*. Chapman and Hall/CRC, 2014.

[10] Hurvich, C. M., and Tsai, C.-L. Regression and time series model selection in small samples. *Biometrika 76*, 2 (1989), 297–307.

[11] McCarthy, L. p5. js. *URL https://p5js. org* (2015).

[12] Rasmussen, C. E. Gaussian processes in machine learning. In *Advanced lectures on machine learning*. Springer, 2004, pp. 63–71.

[13] Remes, S., Heinonen, M., and Kaski, S. Non-stationary spectral kernels. In *Advances in Neural Information Processing Systems* (2017), pp. 4642–4651.

[14] Riihimäki, J., and Vehtari, A. Gaussian processes with monotonicity information. In *Proceedings of the Thirteenth International Conference on Artificial Intelligence and Statistics* (2010), pp. 645–652.

[15] Robert Trangucci, Michael Betancourt, A. V. Prior formulation for Gaussian process hyperparameters. In *Workshop on Bayesian Nonparametrics* (2016), NIPS.

[16] Salvatier, J., Wiecki, T. V., and Fonnesbeck, C. Probabilistic programming in python using pymc3. *PeerJ Computer Science 2* (2016), e55.

[17] Schulz, E., Tenenbaum, J. B., Duvenaud, D., Speekenbrink, M., and Gershman, S. J. Compositional inductive biases in function learning. *Cognitive Psychology 99* (2017), 44–79.

[18] Schwarz, G., et al. Estimating the dimension of a model. *The Annals of Statistics 6*, 2 (1978), 461–464.

[19] Sun, S., Zhang, G., Wang, C., Zeng, W., Li, J., and Grosse, R. Differentiable compositional kernel learning for Gaussian processes. *arXiv preprint arXiv:1806.04326* (2018).

[20] Vehtari, A., Gelman, A., and Gabry, J. Efficient implementation of leave-one-out cross-validation and WAIC for evaluating fitted Bayesian models. *arXiv preprint arXiv:1507.04544* (2015).

[21] Vrieze, S. I. Model selection and psychological theory: a discussion of the differences between the Akaike information criterion (AIC) and the Bayesian information criterion (BIC). *Psychological methods 17*, 2 (2012), 228.

[22] Wilson, A., and Adams, R. Gaussian process kernels for pattern discovery and extrapolation. In *International Conference on Machine Learning* (2013), pp. 1067–1075.

[23] Wit, E., Heuvel, E. v. d., and Romeijn, J.-W. 'All models are wrong...': an introduction to model uncertainty. *Statistica Neerlandica 66*, 3 (2012), 217–236.

Appendix A

Appendix

User Study Results

Figure A.1 shows the drawings of different users in user study. Average rank and extrapolation error for each user are presented in the tables below.

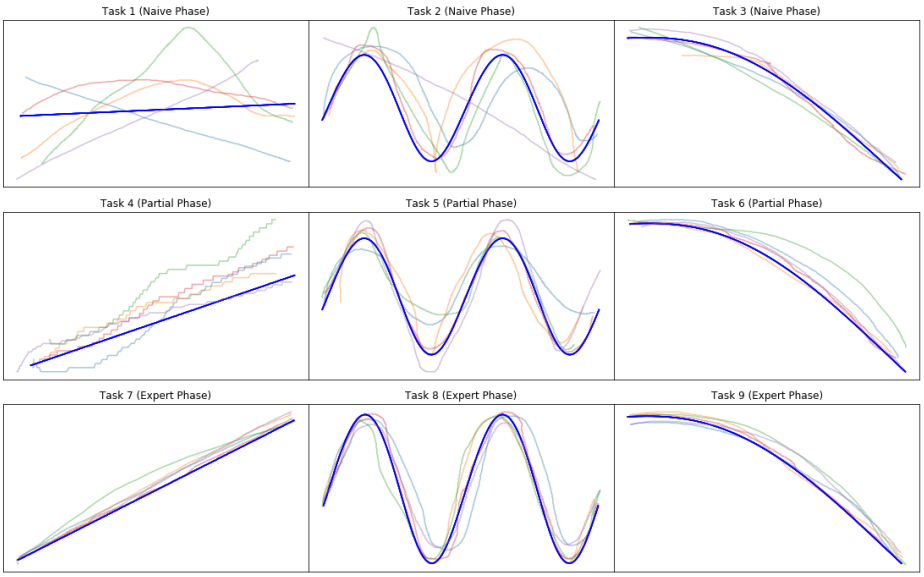

Figure A.1: User drawings from each task and the true function shown in blue. Figures might look different due to the variation in drawing.

Kernel	No of Obs	Average Rank			
		Baseline Model	Naive Phase	Partial Phase	Expert Phase
Linear	16	2.20 (0.87)	2.45 (0.74)	2.50 (0.67)	2.40 (0.80)
	32	2.45 (0.80)	2.50 (0.74)	**2.45** (0.74)	2.70 (0.46)
	64	2.40 (0.80)	2.45 (0.80)	2.60 (0.73)	2.60 (0.66)
	128	2.20 (0.98)	2.40 (0.86)	2.55 (0.67)	2.55 (0.59)
	256	2.30 (0.90)	2.60 (0.58)	2.45 (0.80)	2.50 (0.74)
Periodic	16	1.40 (0.49)	1.50 (0.50)	1.45 (0.50)	1.45 (0.50)
	32	1.05 (0.22)	1.15 (0.36)	**1.00** (0.00)	1.10 (0.30)
	64	1.30 (0.46)	**1.00** (0.00)	**1.10** (0.30)	**1.00** (0.00)
	128	1.35 (0.48)	**1.10** (0.30)	**1.05** (0.22)	**1.00** (0.00)
	256	1.65 (0.48)	**1.15** (0.36)	**1.05** (0.22)	**1.00** (0.00)
RBF	16	1.00 (0.00)	1.05 (0.22)	1.05 (0.22)	1.05 (0.22)
	32	1.00 (0.00)	**1.00** (0.00)	1.10 (0.30)	**1.00** (0.00)
	64	1.00 (0.00)	**1.00** (0.00)	**1.00** (0.00)	1.05 (0.22)
	128	1.00 (0.00)	**1.00** (0.00)	**1.00** (0.00)	**1.00** (0.00)
	256	1.00 (0.00)	**1.00** (0.00)	**1.00** (0.00)	**1.00** (0.00)

Table A.1: Average Rank for User 1.

Kernel	No of Obs	Extrapolation Error			
		Baseline Model	Naive Phase	Partial Phase	Expert Phase
Linear	16	0.26 (0.18)	**0.14** (0.10)	**0.15** (0.14)	**0.15** (0.14)
	32	0.22 (0.11)	**0.15** (0.08)	**0.15** (0.05)	**0.14** (0.05)
	64	0.16 (0.09)	**0.14** (0.05)	**0.13** (0.05)	**0.11** (0.04)
	128	0.11 (0.06)	**0.11** (0.04)	**0.11** (0.04)	**0.11** (0.04)
	256	0.10 (0.04)	0.11 (0.05)	**0.10** (0.04)	0.11 (0.05)
Periodic	16	0.63 (0.20)	0.65 (0.16)	**0.62** (0.18)	0.65 (0.16)
	32	0.29 (0.20)	0.38 (0.30)	**0.26** (0.24)	0.33 (0.27)
	64	0.39 (0.37)	**0.27** (0.22)	**0.22** (0.16)	**0.18** (0.11)
	128	0.35 (0.45)	**0.26** (0.34)	**0.15** (0.10)	**0.14** (0.09)
	256	0.42 (0.37)	**0.18** (0.24)	**0.11** (0.08)	**0.09** (0.04)
RBF	16	0.63 (0.50)	0.86 (1.07)	0.88 (1.18)	0.83 (1.10)
	32	0.40 (0.33)	**0.35** (0.26)	0.95 (1.60)	**0.38** (0.31)
	64	0.27 (0.20)	**0.25** (0.17)	**0.27** (0.20)	0.51 (1.08)
	128	0.28 (0.33)	**0.20** (0.15)	**0.23** (0.17)	**0.22** (0.16)
	256	0.23 (0.14)	**0.22** (0.12)	**0.22** (0.13)	**0.21** (0.13)

Table A.2: Extrapolation Error for User 1.

Kernel	No of Obs	Average Rank			
		Baseline Model	Naive Phase	Partial Phase	Expert Phase
Linear	16	2.20 (0.81)	2.45 (0.80)	2.60 (0.66)	2.50 (0.74)
	32	2.40 (0.73)	2.75 (0.54)	2.65 (0.57)	2.65 (0.65)
	64	2.40 (0.80)	2.55 (0.67)	**2.40** (0.73)	2.50 (0.67)
	128	2.30 (0.90)	2.35 (0.91)	2.50 (0.74)	2.50 (0.81)
	256	2.65 (0.65)	**2.60** (0.58)	2.75 (0.54)	**2.65** (0.57)
Periodic	16	1.40 (0.49)	1.65 (0.48)	1.55 (0.50)	**1.25** (0.43)
	32	1.15 (0.36)	**1.10** (0.30)	**1.10** (0.30)	**1.10** (0.30)
	64	1.30 (0.46)	**1.00** (0.00)	**1.05** (0.22)	**1.00** (0.00)
	128	1.30 (0.46)	**1.10** (0.30)	**1.00** (0.00)	**1.10** (0.30)
	256	1.50 (0.50)	**1.30** (0.46)	**1.15** (0.36)	**1.25** (0.43)
RBF	16	1.00 (0.00)	1.05 (0.22)	1.15 (0.36)	1.10 (0.30)
	32	1.00 (0.00)	**1.00** (0.00)	1.10 (0.30)	1.05 (0.22)
	64	1.00 (0.00)	**1.00** (0.00)	**1.00** (0.00)	**1.00** (0.00)
	128	1.00 (0.00)	**1.00** (0.00)	**1.00** (0.00)	**1.00** (0.00)
	256	1.00 (0.00)	**1.00** (0.00)	**1.00** (0.00)	**1.00** (0.00)

Table A.3: Average Rank for User 2.

Kernel	No of Obs	Extrapolation Error			
		Baseline Model	Naive Phase	Partial Phase	Expert Phase
Linear	16	0.21 (0.14)	**0.15** (0.08)	**0.16** (0.08)	**0.15** (0.08)
	32	0.13 (0.06)	**0.12** (0.06)	**0.12** (0.05)	**0.12** (0.06)
	64	0.13 (0.09)	**0.11** (0.05)	**0.11** (0.06)	**0.11** (0.05)
	128	0.11 (0.08)	**0.10** (0.05)	**0.10** (0.04)	**0.09** (0.05)
	256	0.11 (0.04)	**0.11** (0.04)	**0.11** (0.04)	**0.11** (0.04)
Periodic	16	0.67 (0.29)	0.72 (0.21)	0.69 (0.24)	0.71 (0.23)
	32	0.35 (0.39)	0.42 (0.30)	**0.34** (0.24)	0.48 (0.30)
	64	0.50 (0.75)	**0.22** (0.13)	**0.27** (0.38)	**0.19** (0.07)
	128	0.32 (0.48)	**0.15** (0.12)	**0.12** (0.05)	**0.15** (0.13)
	256	0.41 (0.57)	**0.17** (0.15)	**0.16** (0.18)	**0.24** (0.34)
RBF	16	0.45 (0.30)	0.64 (1.08)	1.12 (1.66)	0.86 (1.38)
	32	0.52 (0.29)	**0.49** (0.27)	0.94 (1.35)	**0.49** (0.27)
	64	0.28 (0.16)	**0.25** (0.13)	**0.25** (0.13)	**0.26** (0.13)
	128	0.20 (0.18)	**0.19** (0.17)	**0.19** (0.18)	**0.20** (0.18)
	256	0.21 (0.16)	**0.20** (0.15)	**0.20** (0.15)	**0.20** (0.15)

Table A.4: Extrapolation Error for User 2.

Kernel	No of Obs	Average Rank			
		Baseline Model	Naive Phase	Partial Phase	Expert Phase
Linear	16	2.30 (0.71)	**2.30** (0.78)	2.40 (0.73)	2.45 (0.74)
	32	2.45 (0.67)	2.65 (0.57)	**2.45** (0.80)	2.60 (0.66)
	64	2.45 (0.86)	2.55 (0.67)	**2.35** (0.91)	2.55 (0.74)
	128	2.55 (0.80)	2.65 (0.65)	**2.50** (0.81)	**2.50** (0.87)
	256	2.30 (0.78)	2.55 (0.74)	2.55 (0.67)	2.55 (0.74)
Periodic	16	1.45 (0.50)	**1.05** (0.22)	**1.20** (0.40)	**1.45** (0.50)
	32	1.15 (0.36)	**1.00** (0.00)	1.20 (0.40)	1.25 (0.43)
	64	1.40 (0.49)	**1.20** (0.40)	**1.00** (0.00)	**1.10** (0.30)
	128	1.50 (0.50)	**1.00** (0.00)	**1.20** (0.40)	**1.15** (0.36)
	256	1.60 (0.49)	**1.15** (0.36)	**1.00** (0.00)	**1.05** (0.22)
RBF	16	1.05 (0.22)	**1.00** (0.00)	**1.00** (0.00)	1.15 (0.36)
	32	1.00 (0.00)	**1.00** (0.00)	**1.00** (0.00)	**1.00** (0.00)
	64	1.00 (0.00)	**1.00** (0.00)	**1.00** (0.00)	**1.00** (0.00)
	128	1.00 (0.00)	**1.00** (0.00)	1.05 (0.22)	**1.00** (0.00)
	256	1.00 (0.00)	**1.00** (0.00)	**1.00** (0.00)	**1.00** (0.00)

Table A.5: Average Rank for User 3.

Kernel	No of Obs	Extrapolation Error			
		Baseline Model	Naive Phase	Partial Phase	Expert Phase
Linear	16	0.27 (0.15)	**0.19** (0.11)	**0.16** (0.10)	**0.15** (0.08)
	32	0.23 (0.10)	**0.15** (0.04)	**0.13** (0.05)	**0.14** (0.04)
	64	0.18 (0.10)	**0.12** (0.04)	**0.15** (0.06)	**0.15** (0.05)
	128	0.09 (0.05)	**0.09** (0.05)	**0.09** (0.04)	**0.09** (0.05)
	256	0.11 (0.05)	**0.10** (0.04)	**0.10** (0.04)	**0.11** (0.04)
Periodic	16	0.63 (0.34)	**0.61** (0.26)	0.68 (0.21)	0.67 (0.21)
	32	0.77 (1.41)	**0.35** (0.25)	**0.45** (0.28)	**0.41** (0.34)
	64	0.25 (0.20)	**0.23** (0.32)	**0.15** (0.08)	**0.21** (0.15)
	128	0.40 (0.42)	**0.10** (0.04)	**0.15** (0.20)	**0.27** (0.51)
	256	0.42 (0.36)	**0.13** (0.13)	**0.09** (0.04)	**0.11** (0.12)
RBF	16	0.69 (0.88)	**0.51** (0.36)	**0.55** (0.40)	1.23 (1.71)
	32	0.43 (0.33)	**0.36** (0.28)	**0.39** (0.29)	**0.41** (0.30)
	64	0.34 (0.29)	**0.31** (0.27)	**0.32** (0.27)	**0.33** (0.28)
	128	0.26 (0.22)	**0.23** (0.20)	**0.24** (0.20)	**0.25** (0.20)
	256	0.21 (0.17)	**0.20** (0.13)	**0.19** (0.14)	**0.19** (0.15)

Table A.6: Extrapolation Error for User 3.

Kernel	No of Obs	Average Rank			
		Baseline Model	Naive Phase	Partial Phase	Expert Phase
Linear	16	2.25 (0.77)	2.95 (0.22)	2.40 (0.80)	2.40 (0.80)
	32	2.40 (0.80)	2.80 (0.51)	2.55 (0.80)	2.65 (0.73)
	64	2.25 (0.89)	2.75 (0.54)	2.55 (0.59)	2.55 (0.67)
	128	2.30 (0.90)	2.85 (0.36)	2.50 (0.74)	2.80 (0.51)
	256	2.35 (0.85)	2.65 (0.57)	2.50 (0.81)	2.70 (0.64)
Periodic	16	1.40 (0.49)	**1.05** (0.22)	**1.00** (0.00)	**1.05** (0.22)
	32	1.30 (0.46)	**1.00** (0.00)	**1.05** (0.22)	**1.10** (0.30)
	64	1.25 (0.43)	**1.15** (0.36)	**1.05** (0.22)	**1.15** (0.36)
	128	1.25 (0.43)	**1.15** (0.36)	**1.00** (0.00)	**1.10** (0.30)
	256	1.55 (0.50)	**1.05** (0.22)	**1.20** (0.40)	**1.20** (0.40)
RBF	16	1.00 (0.00)	**1.00** (0.00)	1.05 (0.22)	1.05 (0.22)
	32	1.00 (0.00)	**1.00** (0.00)	**1.00** (0.00)	**1.00** (0.00)
	64	1.00 (0.00)	**1.00** (0.00)	**1.00** (0.00)	**1.00** (0.00)
	128	1.00 (0.00)	**1.00** (0.00)	**1.00** (0.00)	**1.00** (0.00)
	256	1.00 (0.00)	**1.00** (0.00)	**1.00** (0.00)	**1.00** (0.00)

Table A.7: Average Rank for User 4.

Kernel	No of Obs	Extrapolation Error			
		Baseline Model	Naive Phase	Partial Phase	Expert Phase
Linear	16	0.31 (0.20)	**0.10** (0.06)	**0.15** (0.13)	**0.13** (0.06)
	32	0.20 (0.14)	**0.13** (0.08)	**0.12** (0.06)	**0.12** (0.06)
	64	0.15 (0.07)	**0.11** (0.06)	**0.12** (0.04)	**0.12** (0.03)
	128	0.12 (0.07)	**0.10** (0.04)	**0.11** (0.04)	**0.11** (0.04)
	256	0.10 (0.04)	**0.10** (0.04)	**0.10** (0.04)	0.11 (0.04)
Periodic	16	0.71 (0.37)	**0.24** (0.11)	**0.41** (0.27)	**0.41** (0.28)
	32	0.77 (1.01)	**0.28** (0.19)	**0.55** (0.77)	**0.36** (0.27)
	64	0.30 (0.36)	**0.21** (0.19)	**0.21** (0.20)	**0.28** (0.40)
	128	0.31 (0.43)	0.33 (0.47)	**0.09** (0.05)	**0.14** (0.17)
	256	0.55 (0.58)	**0.10** (0.07)	**0.19** (0.26)	**0.14** (0.20)
RBF	16	0.57 (0.36)	**0.53** (0.34)	0.72 (0.95)	0.73 (1.02)
	32	0.45 (0.40)	**0.44** (0.38)	**0.43** (0.37)	**0.43** (0.36)
	64	0.27 (0.20)	**0.27** (0.19)	**0.26** (0.19)	**0.27** (0.19)
	128	0.22 (0.14)	0.23 (0.13)	**0.22** (0.13)	**0.22** (0.13)
	256	0.22 (0.17)	**0.19** (0.14)	**0.19** (0.15)	**0.19** (0.14)

Table A.8: Extrapolation Error for User 4.

Kernel	No of Obs	Average Rank			
		Baseline Model	Naive Phase	Partial Phase	Expert Phase
Linear	16	2.15 (0.85)	**2.05** (0.80)	**2.00** (0.77)	2.45 (0.74)
	32	2.60 (0.73)	**2.60** (0.73)	2.80 (0.51)	**2.60** (0.73)
	64	2.50 (0.81)	**2.45** (0.86)	2.70 (0.56)	2.65 (0.65)
	128	2.20 (0.93)	2.40 (0.80)	2.55 (0.74)	2.55 (0.74)
	256	2.30 (0.78)	2.55 (0.67)	2.35 (0.79)	2.55 (0.59)
Periodic	16	1.35 (0.48)	**1.10** (0.30)	1.50 (0.50)	1.45 (0.50)
	32	1.10 (0.30)	**1.10** (0.30)	1.20 (0.40)	1.25 (0.43)
	64	1.40 (0.49)	**1.05** (0.22)	**1.05** (0.22)	**1.00** (0.00)
	128	1.35 (0.48)	**1.05** (0.22)	**1.30** (0.46)	**1.05** (0.22)
	256	1.30 (0.46)	**1.05** (0.22)	**1.05** (0.22)	**1.05** (0.22)
RBF	16	1.00 (0.00)	1.10 (0.30)	1.05 (0.22)	**1.00** (0.00)
	32	1.05 (0.22)	1.10 (0.30)	1.10 (0.30)	**1.05** (0.22)
	64	1.00 (0.00)	1.05 (0.22)	**1.00** (0.00)	1.05 (0.22)
	128	1.00 (0.00)	1.05 (0.22)	**1.00** (0.00)	**1.00** (0.00)
	256	1.00 (0.00)	**1.00** (0.00)	**1.00** (0.00)	**1.00** (0.00)

Table A.9: Average Rank for User 5.

Kernel	No of Obs	Extrapolation Error			
		Baseline Model	Naive Phase	Partial Phase	Expert Phase
Linear	16	0.24 (0.19)	**0.18** (0.18)	**0.15** (0.13)	**0.14** (0.12)
	32	0.22 (0.14)	**0.15** (0.10)	**0.13** (0.06)	**0.15** (0.10)
	64	0.13 (0.08)	**0.09** (0.05)	**0.11** (0.05)	**0.10** (0.05)
	128	0.09 (0.06)	0.10 (0.05)	**0.09** (0.05)	**0.09** (0.05)
	256	0.11 (0.05)	0.12 (0.04)	**0.11** (0.04)	**0.11** (0.03)
Periodic	16	0.73 (0.34)	0.80 (0.29)	**0.73** (0.19)	0.88 (0.40)
	32	0.32 (0.25)	0.54 (0.26)	0.46 (0.46)	0.44 (0.26)
	64	0.69 (0.90)	**0.24** (0.21)	**0.24** (0.25)	**0.21** (0.15)
	128	0.18 (0.14)	**0.16** (0.22)	0.22 (0.26)	**0.13** (0.09)
	256	0.32 (0.41)	**0.13** (0.17)	**0.12** (0.11)	**0.12** (0.14)
RBF	16	0.46 (0.31)	0.81 (1.23)	0.72 (1.03)	**0.46** (0.31)
	32	0.53 (0.41)	0.89 (1.46)	0.98 (1.50)	**0.48** (0.33)
	64	0.29 (0.13)	0.53 (1.13)	**0.28** (0.13)	0.52 (1.07)
	128	0.26 (0.22)	0.51 (1.13)	**0.25** (0.22)	**0.25** (0.22)
	256	0.27 (0.18)	**0.26** (0.16)	**0.26** (0.16)	**0.25** (0.16)

Table A.10: Extrapolation Error for User 5.